Report No. CG-D-16-97

Concept Development and Feasibility Study for an Integrated Aids to Navigation Software System

Warren Heerlein

U.S. Coast Guard
Research and Development Center
1082 Shennecossett Road
Groton, CT 06340-6096

FINAL REPORT
October 1996

This document is available to the U.S. public through the
National Technical Information Service, Springfield, Virginia 22161

Prepared for:

U.S. Department of Transportation
United States Coast Guard
Operations, (G-O)
Washington, DC 20593-0001

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

The contents of this report reflect the views of the Coast Guard Research & Development Center. This report does not constitute a standard, specification, or regulation.

Marc B. Mandler
Technical Director
United States Coast Guard
Research & Development Center
1082 Shennecossett Road
Groton, CT 06340-6096

Technical Report Documentation Page

1. Report No. CG-D-16-97	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Concept Development and Feasibility Study for an Integrated Aids to Navigation Software System		5. Report Date October 1996
		6. Performing Organization Code Project No. 2700
7. Author(s) Warren Heerlein		8. Performing Organization Report No. R&DC 29/96
9. Performing Organization Name and Address U.S. Coast Guard Research and Development Center 1082 Shennecossett Road Groton, Connecticut 06340-6096		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.
		13. Type of Report and Period Covered Final Report
12. Sponsoring Agency Name and Address U.S. Department of Transportation United States Coast Guard Operations, (G-O) Washington, DC 20593-0001		14. Sponsoring Agency Code Commandant (G-OPN) USCG Headquarters Washington, DC 20593-0001

15. Supplementary Notes

The R&D Center's technical point of contact is Warren Heerlein, 860-441-2625.

16. Abstract

This report examines the feasibility of integrating varied, preexisting short-range aids to navigation (AtoN) software. The applications design, implement and manage AtoN for the Coast Guard. It is envisioned their functionality can be recombined into one application to yield open access to AtoN data and provide a systematic approach to engineering and management tasks. Feasibility is addressed with respect to technical risk, management issues and cost.

Following recent trends in client/server applications, arguments are made in favor of an Intranet design. This will minimize client software costs, reduce training and allow for central management of the application. Although the system will operate in vastly different geographic locations, the Intranet concept is still applicable because the wide area network will be private and secure.

The Coast Guard applies research and new technologies, like Intranets, to enhance waterways performance. Improved waterways benefit the nation by increasing commercial mobility, recreational access, maritime safety and environmental protection.

17. Key Words		18. Distribution Statement
aids to navigation (AtoN)	Intranet	Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161
client/server spplication	networks	
Coast Guard	software	
distributed processing		

19. Security Classif. (of this report)	20. SECURITY CLASSIF. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED		

Form DOT F 1700.7 (8/72) Reproduction of form and completed page is authorized

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (WEIGHT)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (EXACT)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly).

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (WEIGHT)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	0.125	cups	c
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (EXACT)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

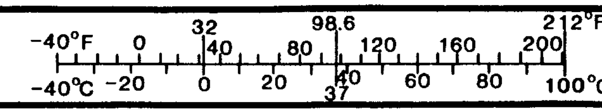

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

EXECUTIVE SUMMARY

Nine software applications are currently being used by Coast Guard personnel to assist in design, implementation and management of aids to navigation (AtoN). The computer programs are relied upon daily to complete various aspects of AtoN work. Some are executed over 100,000 times a year Coast Guard wide.

The applications have met or exceeded a reasonable life expectancy. Some of the programs are over twenty years old and have design deficiencies inherent to that era. One problem with the programs is data quality. The applications duplicate many data-items, but have no facility to update each other. Another problem is data availability. The programs do not provide remote access for users who are not physically connected to the software's host machine. Additionally, analysis has shown the software will not operate on next-generation Coast Guard workstations presently being deployed. A new software system is needed.

The feasibility of integrating the existing programs into an Intranet application is explored in this report. An Intranet uses a Web server to deliver information to users connected across a private network. The concept is technically feasible, but areas of risk have been identified in the bandwidth of the Coast Guard Data Network and the ability of browsers to perform satisfactorily with complex processes like those needed to position or moor AtoN.

Waterway performance could be significantly degraded if new AtoN software is not deployed in a reasonable time-frame or if it is introduced piecemeal. The use of Intranet technology should be given careful consideration because it may be the fastest and least expensive way to deploy a new system.

ACRONYMS AND ABBREVIATIONS

AAPS..........Automated Aid Positioning System

ADAM..........Aid Design and Management

ADP...........Automated Data Processing

ANMS..........Automated Notice to Mariners System

API...........Application Programming Interface

APR...........Aid Positioning Record

ARRF..........Automated Relative Risk Factor

ATON..........Aids to Navigation

ATONIS........Aids to Navigation Information System

CAD...........Computer Aided Design

CAPT..........Captain

CDR...........Commander

CGDN..........Coast Guard Data Network

CGI...........Common Gateway Interface

CGSW..........Coast Guard Standard Workstation

COTS..........Commercial Off The Shelf

CTOS..........Convergent Technologies Operating System

CUA...........Common User Access (IBM®-1989)

DB............Database

DCOM®.........Distributed Component Object Model

DGPS..........Differential Global Positioning System

DDE®..........Dynamic Data Environment

DMA...........Defense Mapping Agency

ACRONYMS AND ABBREVIATIONS (continued)

DOS...........Disk Operating System

DOT...........Department of Transportation

ECDIS.........Electronic Chart Display and Information System

EECEN.........Electronics Engineering Center (USCG)

FIPS..........Federal Information Processing Standards

G-O...........Operations Organization (USCG)

G-S...........Systems Organization (USCG)

GDOC..........Geographical Display Operations Computer

GIS...........Geographical Information System

GOSIP.........Government Open Systems Interconnectivity Profile

GPRA..........Government Performance and Results Act

GPS...........Global Positioning System

GUI...........Graphical User Interface

HQ............Headquarters

HTML..........Hypertext Markup Language

HTTP..........Hypertext Transfer Protocol

IBM®..........International Business Machines Corporation

IMO...........International Maritime Organization

ISAPI.........Internet Server Application Programming Interface

ISO...........International Standards Organization

I/O...........Input/Output

JPL...........Jet Propulsion Lab

LAN...........Local Area Network

ACRONYMS AND ABBREVIATIONS (continued)

LAAPS.........Laptop Automated Aid Positioning System

LNM...........Local Notice to Mariners

LORAN.........Long Range Radio Aid to Navigation

MBS...........Million bits per second

MEA...........Mission Essential Application

MICROSOFT.....Microsoft Corporation

MOORSEL.......Computerized Mooring Selection Guide

MSIS..........Marine Safety Information System

NAVCEN........Navigation Systems Center (USCG)

NIS...........Navigation Information Service

NMEA..........National Marine Electronics Association

NOAA..........National Oceanic and Atmospheric Administration

NOS...........Network Operating System

NSS...........Navigation Safety System

NT®...........Microsoft Windows-NT®

OAB...........Operational Advisory Broadcast

ODBC..........Open Database Connectivity

OLE®..........Object Linking and Embedding

PC............Personal Computer

POSIX.........Portable Operating System Interface Exchange

RAD...........Rapid Application Development

RDBMS.........Relational Database Management System

R&D...........Research and Development

ACRONYMS AND ABBREVIATIONS (continued)

R&DC..........Research and Development Center (USCG)

SOLARCALC.....Solar Calculation Program

SQL...........Structured Query Language

TCP/IP........Transfer Control Protocol/Internet Protocol

USAF..........United States Air Force

USCG..........United States Coast Guard

VTS...........Vessel Traffic Services

WAN...........Wide Area Network

WMSD..........Waterways and Marine Safety Division (USCG R&DC)

WWW...........World Wide Web

4GL...........Fourth Generation Programming Language

PROJECT PURPOSE AND EXECUTION

Purpose of Project Element.

The purpose of this project element is to determine the feasibility of integrating varied, preexisting short-range aids to navigation (AtoN) software. The applications under consideration design, implement and manage AtoN. The computer programs, which were developed between 1975 and 1994, are nearing obsolescence. It is envisioned their functionality can be recombined into one application that will yield open access to AtoN data and provide a systematic approach to engineering and management tasks.

Overall Project Benefit.

The goal of the *Short Range Aids to Navigation Research and Development Project* is to better the efficiency and accuracy of the Coast Guard's short-range AtoN program.[1] The program applies research to enhance the performance of waterways. Improved waterways benefit the nation by increasing commercial mobility, recreational access, maritime safety and environmental protection.

Purpose of Report.

This report documents the feasibility study. The report is a final product. As such it meets the deliverable requirement of the *-Integrated Aids to Navigation Software Element-* of the *Short Range Aids to Navigation Research and Development Project*.[1]

Project Execution.

The project element was executed with a combination of in-house and contracted labor. Research and Development Center (R&DC) personnel provided project management, technical expertise, analysis, cost estimation and report writing. A systems analyst was contracted to study existing data structures and to code prototype software as a demonstration of integrated software techniques. Results of this work are recorded in two informal reports. They are attached as: Appendix A, *Laptop Automated Aid Positioning System (LAAPS) and Aids to Navigation Information System (ATONIS) Integration Feasibility Study;* and Appendix B, *ATONIS - Solar Calc Integration Feasibility Study*.

Investigation of the integrated AtoN software concept was requested by the Coast Guard's Operations Organization, Operations Policy Directorate, Office of Aids to Navigation (G-OPN), Short Range Aids to Navigation Division, Coast Guard Headquarters, Washington, D.C.[1]

STRUCTURE AND USE OF FEASIBILITY STUDY

Elements.

This feasibility study is organized with the following elements:

- identification of need
- requirements to meet need
- preliminary design options
- feasibility analysis
- alternative solutions
- summary discussion

The elements follow general criteria used for evaluation of automated data processing (ADP) proposals.[2]

Use.

The primary use of this report should be to inform short-range AtoN program managers about the practicality of committing funds and/or manpower to integrating existing AtoN software into a new application. In the present budget environment, program managers must take a more pragmatic approach to proposals. Denying ADP projects may be as critical to an organization as approving them. Feasibility studies provide input to this decision-making process.

AIDS TO NAVIGATION SOFTWARE UNDER CONSIDERATION

History.

The Coast Guard began using computer technology in the 1970s. In 1975 R&DC contracted with Jet Propulsion Laboratory (JPL) to create a computer program designed to size solar systems used for powering AtoN signals.[3] In 1978, a pilot development for the Marine Safety Information System (MSIS) was contracted to Battelle Labs of Columbus, Ohio.[4] With the advent of personal computers (pcs), Coast Guard staff at units, groups and headquarters found productive ways to use computer programming. Several successful utilities were coded and used informally within the service.

In the early 1980s, the Coast Guard established a standard computing environment using network technology. Specified hardware and software known as the Coast Guard Standard Workstation (CGSW) was promulgated under the auspices of the old Office of Command, Control, Communication and Technology. Computer programs identified as being mission essential applications (MEAs) were compiled to run under the Convergent Technologies Operating System (CTOS®). MEA software joined standard office-productivity applications installed on CGSW. After a follow-on contract was negotiated years later, CGSW became known as CGSW-II.

The CGSW-II, which is still in use, also offers a portable model which is a laptop computer running Disk Operating System (DOS) compatible software. In general, MEA software does not execute on the portable model.

This report addresses computer systems that serve the short-range AtoN program. The applications are primarily used to design, position and manage AtoN. They are also responsible for recording and reporting marine information.

Software Identified for Integration.

AtoN software identified for integration are:

AAPS............Automated Aid Positioning System
ADAM............Aid Design and Management System
ARRF............Automated Relative Risk Factor Program
ATONIS..........Aids to Navigation Information System
LIGHT LIST......Light List Database
MOORSEL.........Computerized Mooring Selection Guide
NIS.............Navigation Information Service
RANGE DESIGN....Range Design Program
SOLARCALC.......Solar Calculation Program[5]

The following paragraphs briefly describe each application.

AAPS. AAPS is a near real-time software application. It is designed to position or position-check floating aids to navigation.[6] AAPS uses output from differential global positioning system (DGPS) receivers or manual entry of horizontal sextant readings to calculate a position. The software displays a plot and target which operators can maneuver towards prior to placing an aid. The plot is derived by comparing real-time position solutions to a recommended position stored in the software. An Aid Positioning Record (APR) is produced as a report of the positioning evolution. AAPS and ATONIS record many of the same aid attributes that describe position. Examples are: name, latitude and longitude. A detailed analysis is provided in Appendix A.

AAPS is used primarily by AtoN field units. AAPS is designed to run under DOS on laptop computers. It is written in Borland Corporation's Turbo-Pascal® programming language. It is commonly referred to as LAAPS, short for Laptop Automated Aid Positioning System. AAPS is not an MEA.

ADAM. ADAM serves as an engineering and office productivity tool. It is a prototype database application that assists with the calculations required for lighted aid design. In addition to visual signal design, it provides AtoN project management. The design solutions recommend standard lamp and lantern combinations that meet a stated operational requirement. The AtoN project-management module provides record-keeping, status reports and a forms generator. The software has a function that will provide

data to a forms server so automatic printing of standard Coast Guard forms like CG-3213 (AtoN Operational Request) can be accomplished with minimal effort.

ADAM has limited distribution to District offices and Headquarters. Input data used by ADAM are also found in ATONIS. Examples of data are: position, height of optic, lamp and lantern. Like ATONIS, ADAM was developed with Progress Corporation's PROGRESS® relational database system and fourth generation programming language (4GL). Because of this compatibility, ADAM can read data from ATONIS when both systems are properly configured. ADAM is not an MEA.

ARRF. ARRF is a computer program that accompanies the Waterway Design Manual.[7] The Waterway Design Manual supplements the Aids to Navigation Manual - Administration (COMDTINST M16500.7), Chapter 4, which documents Coast Guard practices for designing navigational marking systems.

ARRF is designed to analyze AtoN configurations in deep-draft channels used by large commercial vessels. Large displacement vessels (10,000 - 500,000 long tons) are the least maneuverable ships found on waterways. ARRF will calculate the risk of a design vessel straying outside channel boundaries given input about the type of vessel, short-range aids configuration and cross-track current. The calculation is presented as a probability measured from zero to one, with one being certainty. The risk calculations are based on data collected from observing professional mariners who were operating bridge simulators under various scenarios.

ARRF is not an animation system. It is a character-based application. There are no graphics. Users supply data to prompts or make selections from lists. The program is designed to run on CGSW-II. The target user group is the waterways management staff of a District office. ARRF is not an MEA.

ATONIS. ATONIS is a database and application which records features of federal and private AtoN under Coast Guard jurisdiction.[8] It is deployed at Coast Guard Districts and units. Its purpose at the unit level is to provide servicing-schedules and inventories. Examples of data fields are: aid type, hull type, servicing date, lamps, lanterns, etc. At the District level, ATONIS functions primarily to manage aids and process the Local Notice to Mariners (LNM). The LNM reports AtoN deficiencies, chart corrections and other marine safety information to the public.

ATONIS is not a distributed application. Each District has a copy of ATONIS. There is not a central database at Coast Guard Headquarters.

ATONIS uses PROGRESS® software and runs on the CGSW-II. The Coast Guard is currently implementing a new workstation which will use

Microsoft Windows-NT® as an operating system and Microsoft ACCESS® for database. Plans are being made to migrate ATONIS to ACCESS®.[5] ATONIS is an MEA.

LIGHT LIST. The Light List is a publication intended to support interpretation of nautical charts.[9] The Light List provides more information about aids to navigation than can be practically displayed on paper charts. The entire Light List is comprised of seven volumes. Each volume lists aids for a designated geographic area.

The Light List numbers and describes all aids to navigation under Coast Guard authority. Despite the implication of its name, the Light List includes unlighted federal aids and important private aids. Daybeacons and radio aids are examples of unlighted aids. The Light List is formatted in columns. The column headings are:

(1) Light List Number
(2) Aid Name/Location
(3) Position
(4) Signal Characteristic
(5) Height
(6) Signal Range
(7) Structure Type
(8) Remarks

The Light List publications are actually output reports from a database system. The database is maintained by the Defense Mapping Agency (DMA). Light List corrections are forwarded from Coast Guard Districts to Coast Guard Headquarters weekly. DMA provides Headquarters with a terminal link to a system called the Automated Notice to Mariners System (ANMS). Corrections entered at Headquarters update the Light List portion of ANMS at DMA. Only designated individuals with password security can update ANMS. Once a year camera ready copies are provided to Coast Guard Headquarters for printing. The light List is one of the Coast Guard's most important publications. The Coast Guard is required by federal law to maintain and publish it. The LIGHT LIST is not an MEA.

MOORSEL. MOORSEL is a computer program that calculates optimum moorings for buoys.[10] The solution provides sinker size, chain size and length. MOORSEL reads input files that contain attribute data about the aid. These data are similar to that stored in ATONIS and AAPS. Examples are: aid name, position, buoy hull type, depth, etc. After loading data into memory, the program executes algorithms to engineer the mooring load according to expected sea-state conditions. A graphical solution is displayed to the screen.

MOORSEL has been distributed to all AtoN field units. It is written in Borland Corporation's Turbo-Pascal®. It is designed to run under DOS on a pc. MOORSEL is not an MEA.

NIS. NIS is a cost-free public information service.[11] The purpose of NIS is to execute the Coast Guard's charter as the lead public agency for assisting and regulating the civil portion of the Global Positioning System (GPS). NIS is the means for providing civil GPS status reports. NIS is operated by the Coast Guard Navigation Systems Center (NAVCEN), Alexandria, Virginia.

NIS reports diverse radionavigation information and status reports. Examples are: GPS, DGPS, OMEGA, and LORAN. Operational Advisory Broadcasts (OABs) are textual messages that deliver information to radionavigation users. OABs are disseminated through multiple sources. Examples are: Internet, electronic bulletin board, voice recording and radio broadcasts.

In addition to OABs, NIS offers topics such as: U.S. Naval Observatory timing data; USAF/NOAA Solar and Geophysical Activity Reports; NOAA tidal predictions; and, the Coast Guard's LNM.

NIS is not part of the short-range AtoN program. However, NIS is significant because it represents the future direction of how waterways information will likely be provided to users. Conceptually, systems like Light List, ATONIS, and AAPS could output directly to the NIS, providing the most current data available to users. NIS is not an MEA.

RANGE DESIGN. RANGE DESIGN is a computer program used to design two-station navigational range systems.[12] The program calculates the minimum and maximum light intensities allowable at each station. The user is then directed to a luminous intensities manual that translates the values to stock Coast Guard lighting hardware. RANGE DESIGN uses many input variables stored in ATONIS. Examples are: position, dayboard size, structure height, lamp and lantern.

RANGE DESIGN has limited distribution to District offices and Headquarters. These offices typically design and fund high-cost systems like ranges. RANGE DESIGN is designed to run on the CGSW-II. RANGE DESIGN is not an MEA.

SOLARCALC. As previously mentioned, SOLARCALC is one of the Coast Guard's earliest computer programs.[3] Initially developed by JPL, SOLARCALC is a terrestrial version of a program written to size solar-arrays for space vehicles.

SOLARCALC recommends solar-panel and battery capacity for AtoN signals requiring electrical power. SOLARCALC has a file that stores solar insolation data grouped by latitude. SOLARCALC users input latitude, system amperage and battery storage data. This information is also stored in ATONIS. A more detailed study can be found in Appendix B.

SOLARCALC has limited distribution to AtoN units and Districts. The majority of users are found at Districts and Headquarters. SOLARCALC is designed to run on the CGSW-II. SOLARCALC is not an MEA.

IDENTIFYING THE NEED FOR AN INTEGRATED ATON SOFTWARE SYSTEM

Overview.

In general, computer programs designed for the workplace should improve product and/or productivity. AtoN software currently does both. Why then, should the utilities be integrated or reworked? The answer can be found in several events which will be affecting the Coast Guard's near-term future.

The Coast Guard has been streamlining its organization to meet requirements of the Government Performance and Results Act (GPRA).[13] GPRA mandates agencies receiving federal dollars improve their business practices and be more effective in service to the public. In addition to GPRA, the federal government is also making plans to operate with a balanced budget. Agencies will likely suffer significant funding reductions when this occurs. Useful applications, like the proposed integrated AtoN software, may help the Coast Guard offset the effects of possible resource reductions while increasing the efficiency of AtoN commands.

Compounding the concerns over budget limitations is the fact that existing AtoN software's very survival is in question. This is because the next CGSW contract has been finalized and brings with it sweeping changes to the Coast Guard's ADP environment. CGSW-III, the successor to CGSW-II, will employ client/server architecture and consist of local area networks (LANs) of pcs and support devices operating under Microsoft Windows-NT®.[14]

Existing Coast Guard applications must be made compatible for CGSW-III. AtoN programs like AAPS, MOORSEL and unit-level ATONIS, that can operate under DOS, could theoretically run under a DOS partition of Windows-NT®. However, this concept must be tested and verified. Even if the DOS option will work, the programs will not have 32-bit performance. In general, it can be said AtoN software will not operate on CGSW-III because it is unlikely the system will be configured with special features like a DOS partition.

Coast Guard applications approved as an MEA will be migrated to CGSW-III as part of a well planned and funded transition process.[14] Non-MEAs will not be migrated and must be converted by their sponsor programs on an individual basis. As previously stated, almost all AtoN software falls into the non-MEA category. A migration strategy for AtoN software must be developed if the utilities are to have a secure future. Short-range AtoN program managers have speculated that even some loss in capability may result in a degradation of waterways performance.

Seven factors combine to form a reasonable basis for developing an integrated AtoN software system. They are:

- the pending obsolescence of existing software
- the heavy usage of existing software
- the need to align with strategic goals
- the preparation for pending resource reductions
- the need to provide universal access to computerized AtoN design
- the need to improve data integrity, quality and availability
- the need to streamline software maintenance

Pending Obsolescence of Existing AtoN Software.

As mentioned in the overview, existing AtoN software will not operate on CGSW-III. An implementation plan for CGSW-III has been outlined in ALCOAST 069/95.[14] The plan calls for installations to begin at Headquarters in FY96. This will be followed by implementation at Districts and large administrative field units in FY97. The final phase of the plan will install CGSW-III at large operational field units, stations and cutters. Based on ALCOAST 069/95, all CGSW-IIs should be replaced within five years. Using this criterion, existing AtoN software will be obsolete by the year 2000.

Current Usage of Existing AtoN Software.

The following table provides estimates of how often AtoN software is executed. The figures were provided by Coast Guard Headquarters.[5]

APPLICATION USAGE

USERS	ATONIS	AAPS	MOORSEL	SOLAR -CALC	ARRF
Districts (9)	50,000 runs/yr	N/A	500 runs/yr	500 runs/yr	250 runs/yr
AtoN Units (144)	150,000 runs/yr	150,000 runs/yr	15,000 runs/yr	15,000 runs/yr	100 runs/yr

Headquarters, Groups and other activities of the Coast Guard also run these programs. Figures for their use were not provided and are not included in the estimates. NIS runs continuously at the Navigation Systems Center (NAVCEN), Alexandria, Virginia.

It should be noted the importance of software cannot be measured solely by usage. Other factors must be considered. For example, applications like AAPS are especially important because the Coast Guard has adopted radionavigation practices for positioning AtoN. Reverting back to positioning by visual techniques would require

retraining of personnel and significantly reduce unit productivity.

Alignment With Coast Guard Strategic Goals.

The *Commandant's Direction* provides a vision of, and direction for, the Coast Guard's future. Goal eight of the current *Direction* states:

> *"Pursue and exploit new technologies to achieve gains in productivity and enhance mission performance."[15]*

To meet this goal Coast Guard management has committed to:

> *"♦ redirect efforts in Research and Development to further mission productivity.*
> *♦ use technology to enhance maritime safety, surveillance and environmental systems.*
> *♦ be a partner with DOT's R&D efforts to develop integrated smart transportation and navigation information systems.*
> *♦ manage Coast Guard information resources."[15]*

The Coast Guard is stating a need to work smarter and integrated AtoN software can help meet that need. In addition, needs that align with strategic goals should receive adequate attention, funding and leadership support.

Preparation for Resource Reductions.

Recent Congressional and Administration actions aimed at balancing the federal budget are forcing government agencies to prepare for funding curtailments. The Coast Guard will need ideas for ways to maintain its quality services with fewer billets and dollars.

At the same time, waterways use is increasing. United States waters currently serve twenty-six million recreational boaters. Waterways also deliver over one-billion tons of commerce annually.[16] The significance of AtoN cannot be down-played. By marking navigable water, mobility and safety are enhanced. Ports are spared environmental damage that might otherwise occur from certain marine casualties.

In some harbors, sophisticated multi-sensor systems, called Vessel Traffic Services (VTS), provide additional monitoring of vessel movements. However, for the majority of ports and mariners, safe water is simply defined by the visual short-range AtoN marking system.

To be successful in the next millennium, the Coast Guard will need improved methods to manage AtoN. One solution is to provide quality software. This will help AtoN units perform their tasks more efficiently.

Providing Full Access to Computerized Systems Design.

AtoN software would be more effective if an agenda to package, promulgate and practice the individual programs were scheduled. A clearer vision of who should be using the software, and when they should be using it, is needed. The utilities perform adequately in their singular capacities, but do not provide the full advantages of computer-assisted systems design, namely accurate and validated solutions.

AtoN design and servicing should be approached from a systems perspective. Each component of an aid must be functioning to ensure signal delivery. The mooring is as important as the light. AtoN system performance is dependent on individual aid availability. By using software to verify AtoN components are meeting minimum design requirements, a consistent and well-engineered system will be ensured and verified.

A possible configuration for integrating existing applications is:

DESIGN TOOLS	MARINE INFORMATION	POSITIONING
ADAM	LIGHT LIST	AAPS
ARRF	ATONIS/LNM	
SOLARCALC	NIS	
MOORSEL		
RANGE DESIGN		

In the present CGSW-II environment, AtoN software uses several different hardware and software configurations. No one can guarantee a designer or servicing-unit will have all the necessary computers or programs. This promotes inconsistent use. A hypothetical situation could develop where part of an aid is designed by computer, part by manual calculation, and part by "seat-of-the-pants" intuition. For example, if AAPS is used to best position a buoy, but the sinker is selected by some method other than MOORSEL, and the aid moves off-station because of an insufficient mooring, what has the AtoN system or the Coast Guard gained towards efficiency?

Software tools are not meant to replace conventional processes, in this case engineering, but to augment them. They must be used consistently to provide maximum benefit. A "virtual AtoN workstation" will provide functionality for all AtoN duties ranging from proposing an aid, to placing it, to reporting discrepancies. In the CGSW-II environment, AtoN software may not be available for a variety of reasons. With integrated AtoN software, improved access to software tools will lessen the likelihood of AtoN discrepancies caused by poor practices or human error. A new integrated AtoN application will represent a step towards providing systems solutions for waterways designers, maintainers and managers.

Improving Data Integrity, Quality And Availability.

Data should not be duplicated between systems. One obvious reason is the dilemma presented by different values for the same entity. Which one should be used? Where practical, systems design should provide uniqueness for core data elements. This will eliminate duplication and promote data quality. Currently, the identified systems do not interface or share data. The chances for viewing aged or incorrect data are increased. This is because an approved procedure for updating companion programs has not been formulated.

Correctly designed databases are needed to eliminate this situation. Physical databases should not be embedded in applications. They should be maintained within database management systems external to, but accessible from applications. The database systems should conform to open database connectivity (ODBC) standards.

Data entry should be made as efficient as possible. In principle, data should be entered once, - at its source. The user interface should be designed to promote accuracy. Direct sensor input and software techniques like initial values, pick-lists and validation can be used to further enhance data quality.

The short-range AtoN program has not invested in developing a model of its information requirements. Existing software does not support program-wide perspectives and access to source data is not convenient. Data calls, which burden units with extra work, are often the means to examine program processes. Integrated AtoN software will improve information management by centralizing subject databases and making data reachable by all levels and types of Coast Guard personnel.

Streamlining Software Maintenance.

Software requires maintenance to remain reliable and effective. Even the best developments rarely deliver error-free systems. New requirements evolve as a system is used. Because of these factors, a normal life-cycle for software will include several revisions. It must be acknowledged that software will have costs after deployment. Existing AtoN software has not been maintained in this manner. This is primarily because other program priorities have expended resources before revisions can be accomplished.

If existing AtoN software was integrated into one system, it would be a matter of maintaining one system and not nine programs. A secondary benefit would be derived from ensuring changes were consistent across the whole system.

By giving this new system MEA status, funding for distribution, maintenance and support could be calculated into an annual budget line-item. The program would not have to assign billets full-time

to handle software problems or maintenance. The security of knowing a well maintained system is in place would let short-range AtoN managers focus their attention on waterways issues.

INITIAL REQUIREMENTS FOR AN INTEGRATED ATON SOFTWARE SYSTEM

Overview.

The proposed integrated AtoN system will be required to improve the process of designing, managing, implementing and servicing AtoN. Integrated AtoN software must help the Coast Guard meet its strategic goal of using new technologies to improve mission performance.

The system should be deployed as a virtual AtoN workstation. The system must directly benefit users by relieving them of the need to understand and correctly apply complex scientific principles, for example, Allard's Law, which is used in lighted aid design calculations. The system should improve user efficiency by making AtoN design and hardware checks effortless and routine.

The proposed integrated AtoN system will be required to improve the quality and integrity of AtoN data. This will be accomplished by designing and deploying relational database management systems (RDBMS) to supply data to the application. The system may require multiple databases. The databases will be required to meet open database connectivity (ODBC) standards. The database systems must support structured query language (SQL) calls. The database systems must allow multiuser capability. Users should be able to create, retrieve, view, update, archive and delete records. The database systems will be required to control user access through embedded security. The system will be required to provide the Coast Guard with an informational interface for its short-range AtoN mission.

The system must provide a graphical user interface (GUI). It must share data with all bundled software including word processing, spreadsheet, forms, presentation and project management applications.

The proposed system will be required to operate across CGSW-III architecture. The system will be required to provide client and server software. The proposed AtoN system will be required to operate across LANs and wide area networks (WANs). It must be compatible with protocols used on the Coast Guard Data Network (CGDN) and the Internet. As a minimum, the software should be compatible with transfer control protocol/internet protocol (TCP/IP). The system should also provide high-level server protocols like hypertext transfer protocols (HTTP) and be capable of importing and exporting hypertext markup language (HTML) products. The system must work with software drivers for approved CGSW-III network devices like printers, plotters and scanners.

The system must provide for communications with radionavigation devices via serial communications ports.

The system should be capable of output conforming to national and international standards. As a minimum, spatial data standards, like Federal Information Processing Standards 173 (FIPS-173), should be met. In addition, International Maritime Organization (IMO) standards for electronic charting and display information systems (ECDIS) should be addressed. The system should be capable of output conforming to International Standards Organization (ISO) data exchange standard number ISO-8211.

The system must be capable of growth and expansion. Scalability of the system must be available both in the number of users and hardware configurations. The life expectancy of the system should be equal to or greater than that of the CGSW-III contract. The system must be Government Open System Interconnectivity Profile (GOSIP) compliant. The system must be easily maintained. The proposed AtoN system must provide on-line and context sensitive help. System administrator and developer/programmer documentation must accompany the system.

The software should be constructed from models of the functional, business and information requirements of the short-range AtoN program. This is a departure from current systems that are function-oriented only. The functional model should mirror tasks performed in the course of AtoN work. The business model should be a higher level view of AtoN work based on organizational and financial systems considerations. The information model should be a view of how data are created, moved through, and out of the AtoN program. This should include warehousing or archiving of data. Interagency and public-access to appropriate data must also be considered.

Performance Requirements.

The proposed AtoN system's performance will be limited by hardware/software solutions available on the CGSW-III contract. As a rule, client activities routed over a WAN should be serviced quickly, but this will be subject to the nature and scope of the request. As a guideline, wait-times should not exceed five seconds for routine system functionality.

It will be necessary for the application designers to work closely with network specialists. High performance predicted for LANS will not be the same across WANS. In addition, remote access is presently limited to 28.8Mbs at best. The system will inherently have points that limit or restrict data flow.

Compensation must be made for system overhead on components such as routers and switching devices. An integrated AtoN software system will likely require detailed performance analysis, especially for links to afloat units, which will need special attention because of the costs associated with wireless data transmission.

Transactions should be cataloged by type and how much data will be sent by each. A distribution of the frequency of the transactions should be developed. This will provide data to feed modeling of system performance. The WAN will likely require links capable of 1.54Mbs (T1)transmission rates as a minimum.

Software design will also be critical. Processing at the client site may be preferable to server processing, especially if large volumes of data will load the network. Stored procedures for database transactions may be required. Server software for databases will need to accommodate searches of at least 100,000 records plus an undetermined scalability factor for growth. This is based on estimates of 50,000 federal and private aids, respectively.

Functional Requirements.

The proposed integrated AtoN system will be required to provide the following modes:

- waterways display
- engineering
- positioning
- aid management
- risk assessment

Waterways Display Mode. The software must incorporate a geographical information system (GIS). GIS technology allows large quantities of diverse data to be displayed in a logical manner. The application of GIS technology to waterways data is necessary and long overdue. Because AtoN is geographically referenced, it is a natural candidate for a GIS application.

The user should be able to view all waterways of interest. The user must have control over display configurations, removing or adding objects as desired. The user should be able to access additional data windows about objects through a point and click interface.

Engineering Mode. This subsystem will be required to engineer:

- visual signals
- mooring systems
- signal power systems
- range systems

The engineering subsystem will prompt users to enter data needed to solve the task at hand. If the user is analyzing an existing aid, the system will be capable of loading all known information automatically. The engineering subsystem will be capable of offering solutions in both physical measurements and stock Coast Guard hardware. The subsystem will be capable of reading databases for visibility, insolation, sea-states and other applicable environmental data. As a minimum the software will

provide all functionality currently residing in SOLARCALC, MOORSEL, RANGE DESIGN and ADAM.

Positioning Mode. This subsystem will be required to provide positioning software. The subsystem must be capable of interfacing with various radionavigation hardware employing standard National Marine Electronics Association (NMEA) interfaces and data messages. This will include, but not be limited to, Global Positioning System (GPS) and Differential Global Positioning Systems (DGPS) receivers.

The requirement to be capable of manual entry of sextant measurements for non-automated positioning evolutions cannot be established in this study. The functionality for sextant data entry makes up a large portion of the total source code written for AAPS 3.1. It may not be practical to maintain this seldom used and costly capability.

The subsystem will be capable of exchanging data with external systems like the Navigation Information Service (NIS). As a minimum, the subsystem will have all radionavigation functionality currently in AAPS.

Aid Management Mode. This subsystem will be required to provide:

- federal and private aid records
- federal and private aid histories
- federal and private aid pending projects
- inventory capabilities
- servicing and positioning reports
- ad-hoc query and reporting

The management subsystem will track and record information on federal and class-one private aids. The software will provide servicing data from a historical perspective. The system will provide engineering trend analysis. As a minimum, the system will provide all functionality currently found in ATONIS and the project management portion of ADAM.

Risk Assesment Mode. The risk subsystem should, as a minimum, provide the functionality found in ARRF. The subsystem will allow users to calculate the probability of a vessel deviating from a defined channel. The system should utilize the GIS display. The user should be able to rearrange aid configurations on the graphical display and calculate the risk of transit using the new arrangement.

PRELIMINARY DESIGN OF AN INTEGRATED ATON SOFTWARE SYSTEM

Design Considerations.

The first consideration of a design should be to identify the infrastructure that will support the system. Integrated AtoN

software should be built to utilize the client/server architecture of CGSW-III. The software should be planned and have clearly defined client and server components. The design should maximize network resources. This will take advantage of the differences in client and server performance specifications. For example, RDBMS will run on platforms having high clock speeds and large memories. Client software will run on less expensive hardware.

When operating across WANs, the system's operation should appear seamless to users. The program should look as if it were being executed entirely on the user's processor. If necessary, an agent may be designed between the client and server to handle extended processing tasks such as complex database transactions.

The second consideration of a design should be to plan the life-cycle of the product. Most organizations recognize that quality software will be costly. What is problematic is that steady offerings of new software and hardware tend to result in constant system upgrades. The situation provides little return on investment.

Today, commercial-off-the-shelf (COTS) software is following a trend away from *WinTel*, *shrink-wrapped*, single-processor applications. Network based solutions are now dominating the market. One reason network solutions are so popular is because software can be delivered to multiple users from a single server. This reduces installation and training costs.

Preferably, client software will be standardized and inexpensive. In the optimum model, client software simply becomes a World Wide Web (WWW) browser and the server becomes a HTTP capable host. In this design, all system software resides on the server. Upgrades are easily accomplished helping extend system life.

Until recently, a disadvantage of this design was found in the limited power of Internet server software. This is changing. COTS Internet servers are now being marketed with the added-value of programming interfaces and multiple communications protocols. Examples are Microsoft's Internet Information Server® (IIS), Netscape Corporation's SuiteSpot Server® and Process Software Corporation's Purveyor® Web Server. The flexability offered in these packages reduces code writing required by developers. Many functions that would normally require custom programming can now be configured through utilities available in the software. One example is the ability to interface with databases. Users can now query data or build reports directly from their browsers when a system is properly configured.

The integrated AtoN server design should include a software layer for data distribution. Client access to subject databases needs to be part of the design. Distributing data will allow the system to be flatter across the user-base. This is a change from the existing vertical or stovepipe Coast Guard applications that presently trap data on local nodes at Districts and units.

The third consideration of a design should be to meet special needs. If custom programming is required to meet functional requirements, the programming interface will allow designers to utilize logic available from several environments. Examples are: Microsoft's Visual Basic® or Visual C++®, Sun Microsystems' Java®, Borland Corporation's IntraBuilder® or Netscape's JavaScript®. This capability will allow the browser to become interactive with the server.

The system design should also utilize the Object Linking and Embedding (OLE®) power available between Microsoft applications. OLE® enables software to link objects through application interfaces. OLE® is an architecture that supports and controls software objects. The power of objects lies in the ability to build libraries of components that can be reused. Microsoft Office® has been selected as the bundled office productivity software for CGSW-III. It is critical to have Coast Guard business applications share information with these products.

It may also be appropriate to consider other protocols like Microsoft's Dynamic Data Exchange (DDE®) or Microsoft's Distributed Component Object Model (DCOM®). DDE® allows users to open multiple application windows (i.e. word processing, spreadsheet, etc.) and drag and drop data between them. DCOM® is a protocol that allows software objects to communicate directly over a network. DCOM will effectively enhance OLE® and in the future both will be known as ActiveX Controls®.

Choosing the Intranet Concept for a Preliminary Design.

If the CGSW-III system architecture best uses client/server applications, then the preliminary design for integrated AtoN software system can initially be resolved by making two critical choices, the network infrastructure and the form and extent of client software. This reduces design options to three:

- custom client/private network (distributed application)
- standard client/public network (Internet)
- standard client/private network (Intranet)

Integrated AtoN software may need custom client software because of the complexity of its requirements, especially the AAPS requirements. However, this choice should be avoided because of the costs associated with maintaining multiple client sites. AtoN units currently number 149. Each modification of the application using this option will require a fresh installtion at each site. Administration and maintenance costs will be very high.

One alternative is to use Internet technology. By deploying the application on an Internet server, the client software can be

reduced to a commercial browser package. This option can offer significant cost savings. Microsoft currently provides its Explorer® browser free with Windows-NT®. If Explorer® is found to meet the requirements of the system, the Coast Guard could effectively have its integrated AtoN user's software at no cost.

The disadvantage to this solution is the system's server software becomes more complex. A second concern is system performance when network traffic becomes heavy. A third problem is the need for security of data and possibly the need to use encryption.

A final and best option is to privatize the system's network by using the CGDN. This eliminates security concerns and allows for specification of system network needs. When a system's network is exclusive, and the system makes use of Internet technology, the term "Intranet" can be applied to the design. Integrated AtoN software should be designed to utilize the CGDN and operate as an Intranet. The advantages of an Intranet design are:

- low client software costs
- ownership and security
- single-site maintenance strategy
- scalability in users and hardware
- ease of distribution

The disadvantages of an Intranet are:

- dependence on server availability
- limitations of network's bandwidth
- complexity of server software
- proprietary nature of server software

Figure 1 shows a simple schematic of an integrated AtoN software Intranet.

Internal Interfaces - Databases, Functional Modes.

The integrated AtoN Intranet software system design must provide for all functional requirements and subject databases. The databases should be designed from functional, business and information process models. Intermodal transportation data of other government agencies should also be considered for inclusion.

The detailed design of database access will need to be carefully addressed. The common gateway interface (CGI), available in all WWW servers, is one option. Users enter requests for data into a "form" on their web page. The request is sent over the network and processed by the CGI portion of the web server software. A script is constructed by the software, loaded into memory and executed. Typically, this will be an SQL query. When the data is extracted, the results are formatted into HTML and returned to the client.

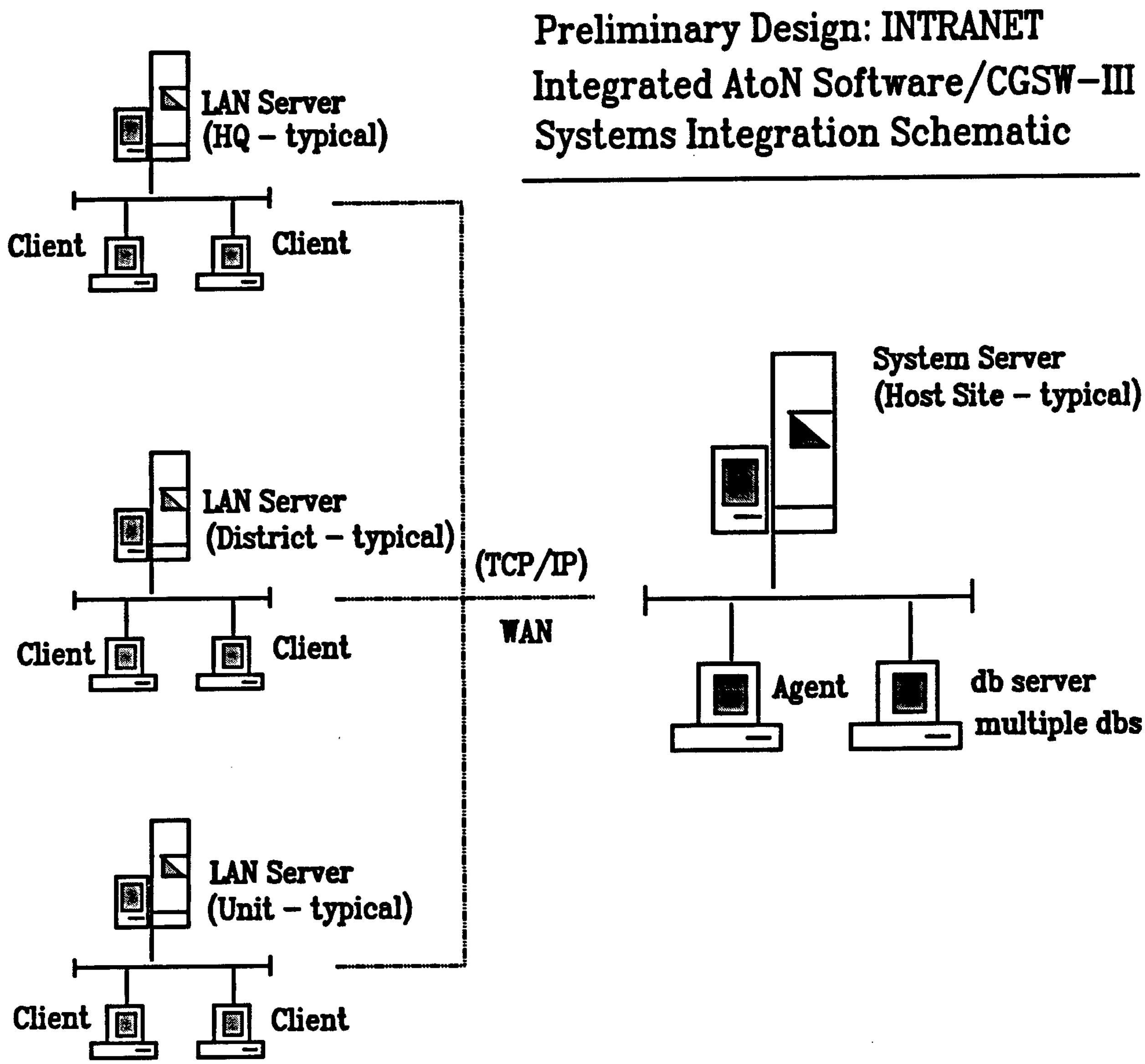

Figure 1. System Schematic

An alternative to CGI is the Internet Server Application Programming Interface (ISAPI). This is an emerging industry standard that allows for more efficient links to applications and databases. The links are programmed through data link library (dll) files. ISAPI is reportedly more efficient than CGI in terms of system memory resources and execution speed.

Figure 2 is a block diagram of some of the system software segments that will be required. The functional blocks are preliminary. The subject databases are examples. Neither are intended to define a detailed design.

Preliminary Design – Integrated AtoN Software

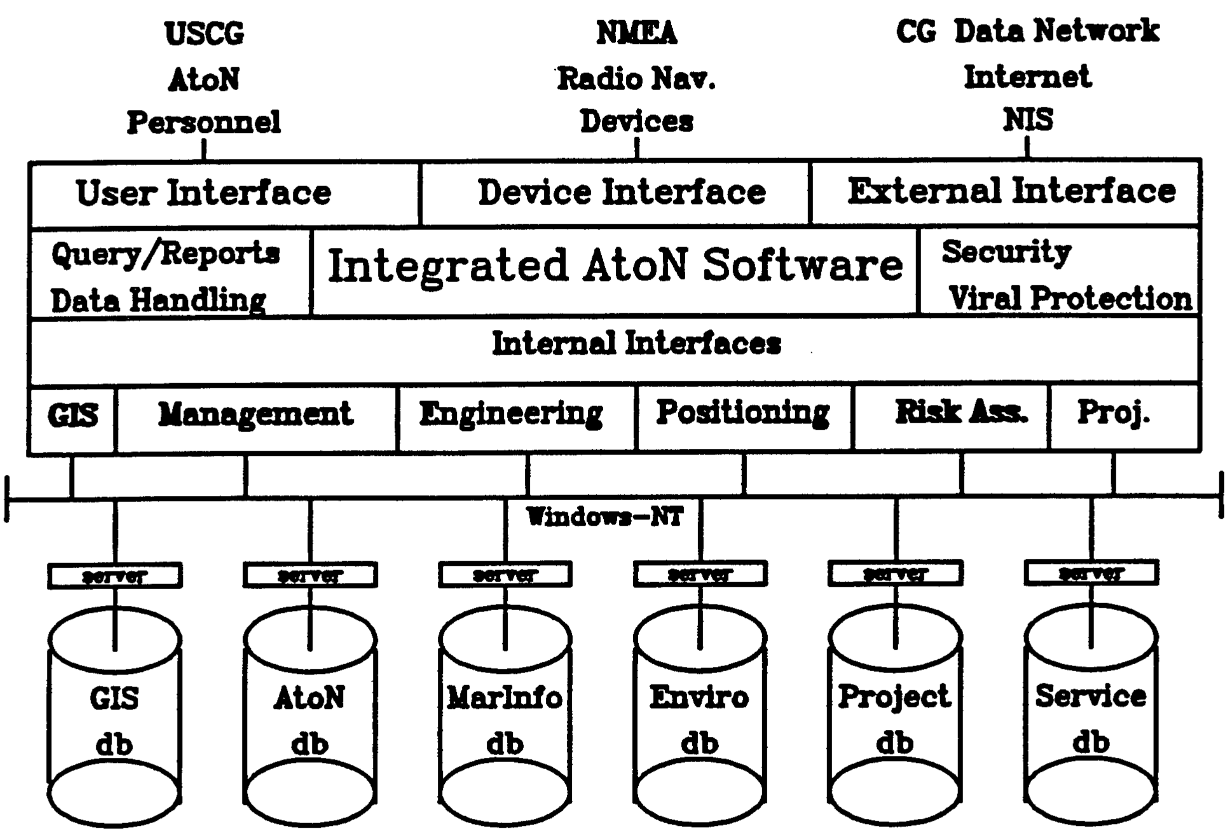

Figure 2. Preliminary Design

External Interfaces - User, Device, Networks.

The system design must include external interfaces for users, radionavigation devices and specified networks and services like NIS. The interface for external radio devices should be modeled on the existing AAPS (version 3.1) design.

The user interface should be provided through a WWW browser. The "home-page" should reflect system requirements as a minimum. In a broader sense, it should be the base of all short-range AtoN information. Manuals, instructions and current events could be hosted there. The page could look like Figure 3, but will be dependent on the browser chosen.

Selections made from the "AtoN Home-Page" should invoke a system server response. Events should appear in standardized windows utilizing GUI concepts. The user interface should be consistent

across all application modules. It should be intuitive to use. The user interface should be based on industry standards, for example IBM®'s Common User Access (CUA) guidelines of 1989. Functionality should be provided by "pointing and clicking" to icons, or as an alternative, chosen from pull-down menus.

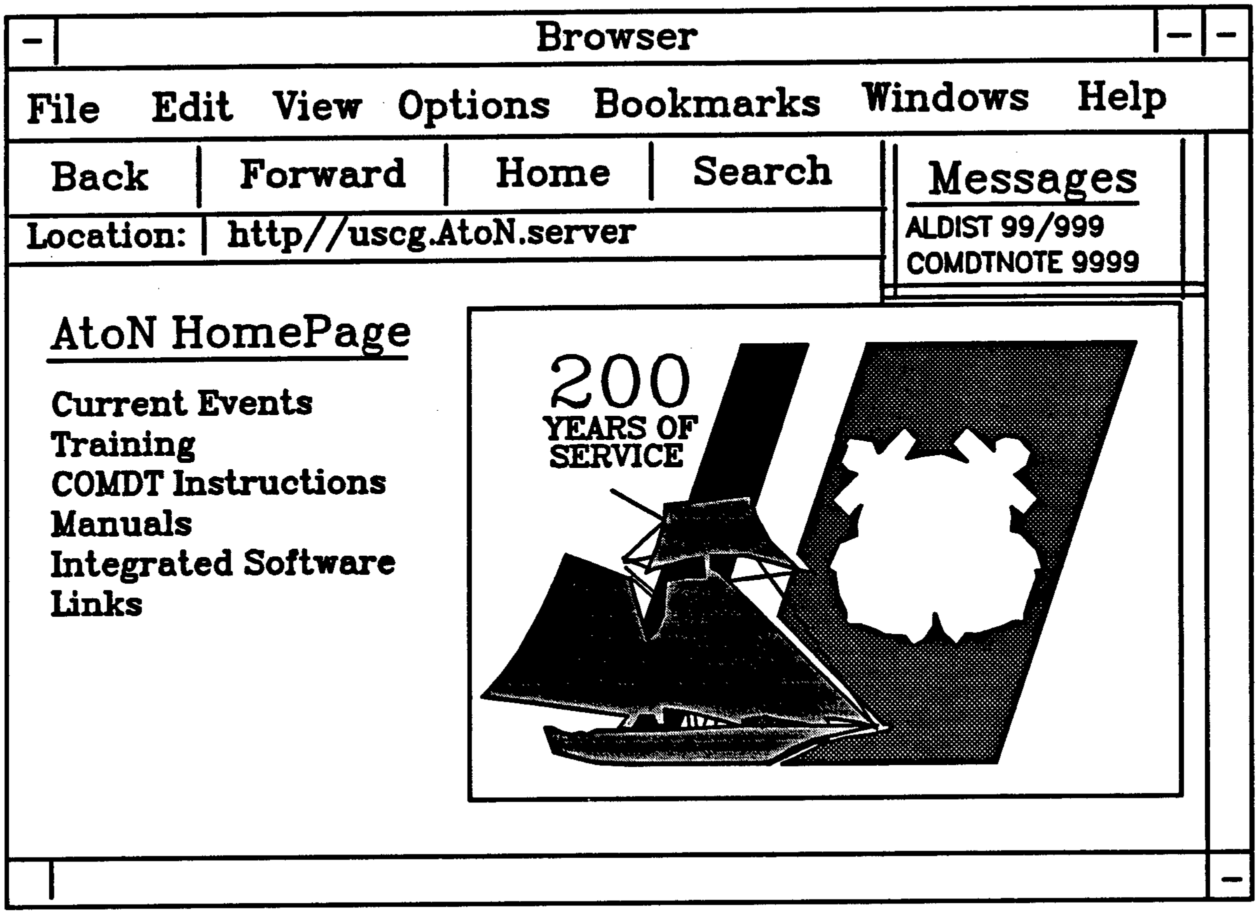

Figure 3. AtoN Home Page

FEASIBILITY OF DEVELOPING AN INTEGRATED ATON INTRANET SOFTWARE SYSTEM

Overview.

It is unrealistic to assume a proposal to develop software will necessarily translate into a useable product. Many things can happen between concept and deployment. It is important to address potential problem areas early-on.

The decision to proceed with integrating AtoN software should be based on overall risk. Cost should not be the only criterion.

Even generous funding can quickly be exhausted when technical or management problems are encountered. Identifying risk in a feasibility study is important because it focuses attention on perceived "weak spots" and factors them into the decision-making process used to approve or disapprove a project.

Three large-issue areas are addressed in this report. They are:

- technical risk
- management issues
- estimated costs

Technical Risk.

The technical risks associated with developing an integrated AtoN Intranet software system will initially be found in four areas. They are:

- development environment
- scalability
- WAN capabilities
- browser capability

Development Environment. System performance will be critical and in some ways dependent on the development environment. Selection of the system server software will likely drive the choice of the development environment. Despite the "open" concept of the Internet, compatibility of system and development software is still an area of concern and risk. Much of the system will likely be configured through utilities provided with the server product. This is a dramatic change from developments where software segments are normally programmed. Examples would be database connectivity and system security. The remainder of the system will probably consist of HTML pages incorporating sophisticated interactive techniques that will require programming with HTML, C or Java® programming languages. Third party object code may also play a role. It will be risky to find the appropriate development tools that best match with the server and client software needed by the system. A server package that includes an application programming interface would best fit the needs of a development.

In proper application design close attention must be paid to requirements gathering and the development of accurate models showing work processes. The models must be mapped to the physical system of software and hardware. Most of this complexity can now be minimized by employing powerful development tools. Examples of products are: Microsoft ActiveX Control Pad®, Moai Technologies' Business Object Architecture (BOA®) or JYACC's Jam-7®. The challenge for designers is to match system requirements to client and server logic in an efficient and cost-effective manner that maximizes performance.

Scalability. Scalability, or the system's capability to grow, is also important. Growth can be expected in the number of users,

quantity of information, and in the power of server software. An integrated AtoN Intranet software system's design should be able to accommodate changes and continue to function properly. Designing for scalability is risky because future needs and developments are hard to forecast. Future changes in system software and network hardware can also impact system performance in unpredictable ways. For example, if a router or other network hardware is added without careful analysis, the system may develop bottlenecks that could crash the system or affect performance.

WAN Capability. This issue presents considerable risk, because the system's design is dependent on communication over WANs. A thorough knowledge of networks is necessary. CGSW-III will be interconnected as LANs using EtherNet media. The LAN servers will be joined into WANs using industry standard Transfer Control Protocol/Internet Protocol (TCP/IP). Windows-NT® provides TCP/IP software protocols.

The CGDN, the Coast Guard's current WAN infrastructure, uses an X.25 transmission protocol. The network is comprised of leased-dedicated lines, dial-up modems and other physical hardware owned by the Coast Guard. X.25 will not provide enough bandwidth for joining CGSW-III LANs and transmitting large volumes of data. The specification of a more robust CGDN will need to be known before completing the system's detail design. CGDN should be upgraded to T1 status which should provide sufficient bandwidth at 1.54 million bits per second (Mbs).[17] Other WAN traffic may require CGDN capacity be increased even further.

The consideration of how afloat units will connect to the network is another design issue having high risk. One option is to use satellite links. Another is cellular voice phone. A third choice may be dedicated radio frequencies using digital data packet technologies.

Any wireless linkage must be carefully analyzed because of the high cost of these services. It may be practical to utilize lower-technology approaches like not designing the system for remote access, but instead employing "usage" strategies. For example, batch updates could be uploaded dock-side prior to a vessels departure. As a minimum requirement, AtoN information for a unit's area of responsibility, must always be available, even if the user is not physically attached to the system. This issue should be resolved as a cooperative effort with Coast Guard personnel responsible for advanced communications systems.

Browser Capability. If a project is elected and the Intranet design is followed, careful consideration must be given to specifying an appropriate browser application. The browser should be capable of compiling Java® source code and executing precompiled programs or objects. Browser-based compilation is a new process and the technology is not mature. Selecting browser software should not simply default to what is available on the

standard bundle. Browser software must meet the requirements of an integrated AtoN Intranet software system. The execution of a GIS using this mode is another area for concern. In general, the further browsers are asked to move away from processing HTML, the higher the risks.

Management Issues.

Management risk associated with developing an integrated AtoN Intranet software system will initially be found in three areas. They are:

- insuring a quality software development plan
- selecting a development methodology
- defining the Coast Guard's project role

Software Development Plan. Because the scope of an integrated AtoN Intranet software project will be large, success will be dependent on adhering to a rigorous software development plan.

The software development plan should fully describe the processes and personnel that will be used to develop the application. The plan should be prepared according to recognized standards.[2] One example is Data Item Description (DID) number DI-MCCR-80030A. This standard outlines the format of a software development plan and includes such items as personnel qualifications, management strategy, development environment, schedules, and product evaluation criterion.

It will be critical to address the system's integration with CGSW-III. Unlike existing single-machine programs, integrated AtoN Intranet software will depend heavily on efficient and reliable operation in a network environment. Explanation of how network operations will be integrated, tested and validated will be an important part of the plan.

Development Methodology. Selecting a development methodology will be another crucial milestone in an integrated AtoN Intranet software project plan. The major challenge will be to deliver a product in a reasonable time-frame. This will depend on specific proposals, but should not be longer than 18 months. In today's market, it is entirely possible for a development to become obsolete if it takes much more than two years to complete.

Risk is associated with choosing how the development will proceed. Traditional development methods, where a system is fully detailed prior to coding, are becoming impractical because they are too time intensive. Large scale developments, like integrated

AtoN Intranet software, are unavoidable and commonplace. Their scope should not pose a barrier to building them. However, new methods for building them should be used. One advantage of the Intranet concept is the ease and speed of setting up a WWW server for testing. Once alpha operations are established, risk issues can be addressed readily. The following paragraphs describe various development methodologies which may be considered.

In <u>prototyping</u>, parts of the system are developed as proof-of-concept. The process typically starts with configuration of test hardware to validate the system's integration with its target architecture. Follow-on work will include mock-ups of the graphical user interface (GUI) with "hooks" for system functionality. As approval for the prototype pieces are granted, the project proceeds with completion of the system's segments. The risk associated with using prototyping is typically schedule and cost related. Accurate task allotments and scheduling is required to avoid problems.

<u>Version release</u> methodology is based on providing an early version of the system that is intentionally missing parts of its functionality. While the release is being used, comments, bug-fixes and enhancements are recorded, often on-line in the software. The next version of the software is then released, which offers fixes of the previous version and additional functionality. This loop continues until the product is completed and deployed. The risk associated with version-release development is when the most complicated parts of the development are left for last. For example, the first release is generally acceptable because it does not have much user-power. The project appears to be on track, but where more complex follow-on versions become bogged-down in technical issues, and the project experiences overruns. An advantage of version release methodology is that it actively involves the user community in the development.

<u>Rapid application development (RAD)</u> is another methodology. RAD is similar to both prototyping and version release and includes elements of both. The distiction of RAD is in requirements gathering and detailed design. The design process is typically carried out within "working groups" of developers, users and management. By bringing these parties together and addressing issues first, a design is quickly agreed on by consensus. Formal coding follows.

One advantage of RAD is that every concerned party has a say and agrees to accept the product as delivered. The risk inherent in RAD is nominating the most appropriate people to the working group. Typically higher-level officials cannot invest the time needed and the product suffers. Another problem-area is that groups "give-up" before the design is reasonable and the final product does not meet anyone's needs.

<u>*Coast Guard's Project Role.*</u> What role the Coast Guard will play in the proposed system's development and life-cycle must be

identified. Will the development be done internally, out-sourced, or be some mix of the two? Clarification of which phases of the development will be done by whom needs to be established.

The Coast Guard has a poor *track* record with regards to software development and management. Many developments have been completed at one activity and then moved to another for implementation and maintenance. The implementation teams are faced with a tremendous learning curve, having not been exposed to the development cycle. The situation is not wrong, but presents another risk which must be addressed prior to a project start.

Another issue is Mission Essential Application (MEA) status. Any software that is being used productively should be considered MEA. The present situation, which has left the short-range AtoN program with only ATONIS as mission-critical, has to be clarified. Program managers need to know their responsibility and authority when considering software development. A clear plan for defining integrated AtoN software as MEA should be incorporated into the software development plan.

Estimated Costs.

The cost of developing an integrated AtoN Intranet software system can be estimated by applying an acceptable metric ($/lines of code, $/function, $/hour, etc.) to required tasks. It may be practical to apply a different metric to different tasks. The final cost estimate can be derived by summing additional costs (i.e WAN, development software, travel, etc.) to the gross development figure.

Cost estimates done for this feasibility study are not formal or rigorous. The intention is to introduce a *ballpark* estimate which the short-range AtoN program can use for internal discussions and long range planning.

Three areas are addressed. They are:

- task allocation
- cost and schedule
- recent experiences

Task Allocation. The allocation of tasks can be broken down as follows:

- Development
  - System Analysis and Specification
    (requirements analysis, interface specifications)
  - Construction and Implementation
    (detail design, code, test, integrate, implement)
- Documentation
  - Technical Writing, Help Subsystem and Illustration
    (users guide, help subsystem, system manuals)

- Publishing
    (reviews, rewrites, master print copies)

♦ Management
- Project Management
    (planning, supervision, risk analysis)
- Reviews & Presentations
    (prototype reviews, demos, travel, hand-offs)

Cost and Schedule. For the purposes of making an initial estimate four assumptions are made. They are:

- the development will be a contracted effort ending with implementation;
- the development will use a hybrid RAD and prototype methodology;
- the system will require no WAN or client costs
- the development will not exceed 18 months

System costs will be attributable to system server software, development software and production software and hardware. A typical project team and their costs including overhead, benefits, general and administrative fees would be:

♦ 1 - Project Manager @($ 70/hr).........	$ 70/hr
♦ 1 - Senior Systems Analyst @($ 56/hr)..	$ 56/hr
♦ 1 - Systems Analyst @($ 49/hr).........	$ 49/hr
♦ 1 - Software Engineer @($ 49/hr).......	$ 49/hr
♦ 1 - GIS Analyst @($ 50/hr).............	$ 50/hr
♦ 1 - Database Administrator @($ 50/hr)..	$ 50/hr
♦ 1 - LAN Specialist @($ 50/hr)..........	$ 50/hr
♦ 3 - Programmers @($ 45/hr).............	$135/hr
♦ 1 - Editor @($ 30/hr)..................	$ 30/hr
♦ 1 - Technical Writer @($ 24/hr)........	$ 24/hr
	=======
	$563/hr[18]

Assuming the full development team will work for 18 months, and using a base of 2,000 hours per working calendar year, the project will require 3,000 contract hours. Multiplying total hours by project team cost per hour produces an estimate of $1,689,000. Additional costs for development hardware and software, office equipment, publishing, presentations, travel and miscellaneous may amount to another $100,000. A full-service Internet server, database servers, HTML production and conversion software costs may add another $100,000. Coast Guard personnel will likely be assigned to the development in a temporary-duty status and are not factored into the costs.

For the purposes of this feasibility study integrated AtoN Intranet software is estimated to cost $2,000,000. This dollar figure represents a first-level estimate only. Figure 4 maps time to tasking.

Recent Experiences. Metrics can often be extrapolated from recent efforts and used to validate initial estimates. ATONIS was

migrated from ReQuest® RDBMS to PROGRESS® RDBMS in 1993. The system was also expanded to include an application for AtoN units. The work was completed at R&DC with a manager, systems analyst and programmer over a two year period. The cost was approximately $300,000. Quality assurance, systems integration and software deployment were done at the Coast Guard's Electronics Engineering Center (EECEN) for approximately $200,000. The total cost for the ATONIS upgrade can be estimated to be $500,000.

ATONIS would represent approximately one-fifth of the proposed modes in the preliminary design. Using this as a scaling factor the estimated cost of the new development would be $2,500,000.

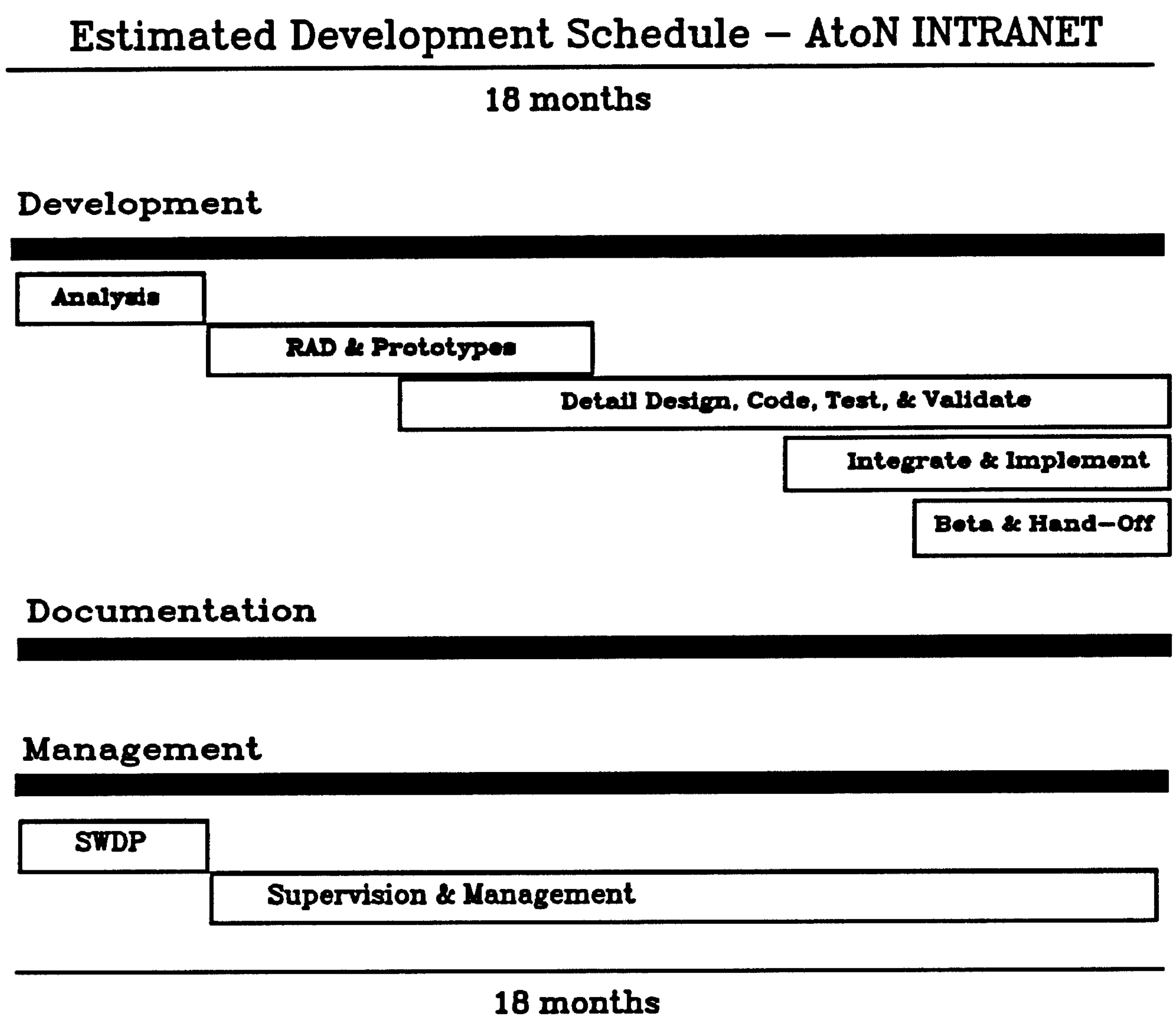

Figure 4. Schedule

ALTERNATIVES TO AN INTEGRATED ATON INTRANET SOFTWARE SYSTEM.

MEA-ATONIS, Version Release Alternative.

One alternative to an Intranet may be to use a version release methodology centered around the MEA conversion of ATONIS to CGSW-III. Software "hooks" could be developed in the ATONIS user interface for all integrated AtoN requirements. Modules of software could then be developed individually over time and offered to users through new versions of ATONIS.

One advantage of this method would be the ability to prioritize module delivery and spread the system cost over several fiscal years. For example, important utilities like AAPS could be integrated sooner. Less important utilities like SOLARCALC could be added in out-years. This approach would allow the short-range AtoN program to take advantage of the "free" conversion of ATONIS to CGSW-III.

The obvious disadvantage of this plan comes from delaying important job aids to the field. Another disadvantage would be the dependence on ATONIS's integration into the CGSW-III architecture. If ATONIS remains decentralized at District offices, the power of client/server architecture will not be used to full advantage.

It is not clear if the conversion of ATONIS will provide for remote data management. Remote data management implies the application will reside on the client machine along with a layer of RDBMS software. The database and transaction software would be installed on the server. This situation, as pointed out in the design chapter, should be avoided because of the high inherent costs of maintaining custom client software. In the long run this will cost the Coast Guard more money.

WAN-less Alternative.

Another alternative would be to use the concept of scalability to meter systems integration. Under this method the system deployment would be done in phases. The system could be deployed to District and unit LANs in a phase I.

When the Coast Guard's WAN is firmly established the system servers located at Districts could be physically moved and reconfigured to a centralized, single-systems server. This evolution would constitute phase II of the project. The advantage of this method is the development could begin without addressing a major project risk factor, WAN capability.

The disadvantage of this method is the risk involved with remapping client software to recognize new locations for the systems server. There is also the chance the move would never be made and the Coast Guard would remain with poorly designed software.

Internet Alternative.

One solution to concerns about the Coast Guard's WAN infrastructure would be to simply use the existing Internet. The advantage of this approach would be the use of a working and proven network. Project risk would be lowered. A second advantage would be that Internet provider costs could be negotiated and controlled. The disadvantages of this plan would be data security, integration and performance of the system on an infrastructure as wide-ranging and open as the Internet.

Alternatives Not Considered, Shell Program.

The concept of a shell for existing systems was not considered to be a viable option. The programs need to be modified to operate in a 32-bit environment. There would be little point in doing this work and not addressing integration and data control problems. Existing software like ATONIS, MOORSEL and AAPS have an overlap of data-fields. A shell for these programs would be inappropriate if data was not normalized between them.

The work presented in Appendix A and Appendix B supports a shell concept. It should be noted this work was performed prior to the award of CGSW-III. In that time, the priority issue was to overcome data duplication between AtoN applications. The use of utilities to transfer data between companion programs was a viable alternative because it was inexpensive compared to revising nine computer applications. With the award of CGSW-III, the emphasis has shifted from a data control problem, to a software obsolescence issue.

SUMMARY DISCUSSION

Feasibility.

Today, it is most important we perform our jobs in an efficient manner. Resources are becoming too scarce to work in any other way. Existing short-range AtoN applications assist those in the field who have access to them. An integrated AtoN Intranet software system should go one step further. It should provide all the software tools needed for AtoN work in a program that is useable by everyone tasked to do those jobs. The technology exists. The tools should be provided.

The short-range AtoN program is long overdue for a program-wide system. An integrated AtoN Intranet software system could enhance program management opportunities by providing an AtoN program information base. Programs contemporaries in the areas of Marine Safety (G-M) and Operations (G-O) have long had similar software tools. The Marine Safety Information System (MSIS) is an example of a system that has been operating for over a decade.

Software development is costly and is growing more so. New Coast Guard systems are being built by other programs. Examples are the Marine Safety Network (MSN) and the Operational Information System (OIS). It may be practical for short-range AtoN program managers to investigate a joint venture with either of these systems. The Ports and Waterways Management Information System (PAWMIS), a module of MSN, could be adapted to short-range AtoN needs. The Coast Guard cannot afford to continually recreate "the wheel" with software systems. Joint ventures between programs should be encouraged and supported.

According to Standish Group International Incorporated, over half of all software development contracts in the United States overrun their estimates. The average overrun is 189%.[19] The Coast Guard cannot afford to be in this half and certainly not multiple times.

The integrated AtoN software concept using an Intranet design is feasible. The risks identified in this report, namely network bandwidth and browser capability, represent development challenges, not roadblocks. Many client/server applications of this type are successfully running in private industry. The Internet has many examples.

This does not imply a client/server application will be easy to build. Coast Guard personnel assigned to the development will need to master software object issues, GUI design, network integration, quality assurance, system performance and development tools. Just because one has the correct architecture does not mean an application running on it will work well. Equally important is the ability to apply development practices and methodologies successfully. Understanding network protocols and standards is critical.

Systems should be designed as much for flexibility as for functionality. When new technologies offer cost-effective advantages they should be able to be integrated into existing systems. You should not have to build a new system with every offering by industry. The Coast Guard should institute software that is open and scalable. GOSIP compliant systems are mandated by law for federal agencies. Despite this, proprietary systems continue to creep into the program. One example is the Electronic Chart Positioning Information and Navigation System (ECPINS) installed on the new WLB buoytender USCGC JUNIPER. ECPINS offers an integrated solution for AtoN work. It has an AtoN database and positioning software combined with an electronic charting and navigation system. ECPINS is a good example of the beginnings of a software system the short-range AtoN program could develop for operational software. However, it is a poor example of how a system should be integrated into the Coast Guard's standard computing environment. It cannot readily communicate with other Coast Guard applications and it is supported with proprietary data.

Other waterways software is also being developed and moving towards prototype testing. The Waterways Evaluation Tool (WET) application and the Integrated Navigation System (INS) pilot project for electronic chart updating are two examples. Both should be investigated for absorbtion by an integrated AtoN Intranet software development.

The short-range AtoN program is not heavily invested in legacy information systems. Some may argue the Light List is legacy, but it could easily be absorbed into a new system. This is a good juncture to venture into a large systems development, when the problems associated with designing to older systems will not be a factor.

Waterway performance could be significantly degraded if new AtoN software is not deployed in a reasonable time-frame or if it is introduced piecemeal. Integrated AtoN Intranet software is a valid concept and technically feasible. The use of Intranet technology should be given careful consideration because it may be the fastest and least expensive way to deploy a new system.

Bibliography

1. USCG R&D Project #2700 - Short Range Aids to Navigation Research - Project Implementation Plan; USCG Research and Development Center, Groton, CT; December, 1994.

2. Developing Software to Government Standards; William H. Roetzheim; Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632; ISBN 0-13-829755-X; 1991.

3. Solarcalc - Solar Design Program Manual; U.S. Coast Guard Headquarters COMDT(G-ECV-3C); November, 1990.

4. An Overview of the Marine Safety Information System; U.S. Coast Guard Headquarters COMDT(G-MP-4); Prepared by Battelle, Columbus Division, Columbus, Ohio; May, 1988.

5. USCG Electronic Mail - Message ID: NSR000115759, between M.Peterson/G-NSR and W.Heerlein/RDC05; May, 1996.

6. Aids to Navigation Manual, Positioning - COMDTINST M16500.1B; U.S. Coast Guard Headquarters COMDT(G-NSR-1); August, 1991.

7. Waterway Design Manual, Report # CG-D-18-92; M.W. Smith; U.S. Coast Guard Research and Development Center; September, 1992.

8. 1995 Five Year Information Resource Management Plan (FYIRMP); United States Coast Guard, Office of Command, Control and Communications, Planning and Architecture (G-TA); 1995.

9. Light List, COMDTPUB P16502.1; Department Of Transportation, U.S. Coast Guard; U.S. Government Printing Office, Washington, D.C.; 1994.

10. MOORSEL - Computer Aided Mooring Selection Guide - Design Specification; U.S. Coast Guard Research and Development Center; October, 1994.

11. U.S. Coast Guard Navigation Information Service (NIS) and Its Function With in the Civil GPS Service (CGS); Commanding Officer, U.S. Coast Guard Navigation Center; Alexandria, Virginia; October, 1994.

12. Aids to Navigation Manual, Administration - COMDTINST M16500.7; U.S. Coast Guard Headquarters COMDT(G-NSR-1); November, 1991.

13. Public Law #103-62; Government Performance and Results Act (GPRA) of 1993; Library of Congress, Washington, D.C.

14. ALCOAST 069/95, Subj: Standard Workstation III Award, Sitrep Two; FROM: COMDT COGARD WASHINGTON DC, TO: ALCOAST; July 1995.

Bibliography (continued)

15. *Commandant's Direction*; United States Coast Guard; Washington, D.C.

16. The United States Waterway System- FACTS; Navigation Data Center, United States Army Corps of Engineers, Alexandria, VA; January 1995.

17. Computer Networks - Protocols, Standards and Interfaces; Uyless Black; Prentice-Hall Inc., Englewood Cliffs, New Jersey 07632; ISBN 0-13-175605-2; 1993.

18. R&DC MFAC Contract for Information Systems Technology, 1996.

19. *PC Week*; March 6, 1995; Volume 12, Number 9; page 21, par. (1).

Laptop Automated Aid Positioning System (LAAPS) and Aids to Navigation Information System (ATONIS) Integration Feasibility Study

Melvin C Wiley
UNISYS Corporation

U.S. Coast Guard Research and Development Center
Avery Point, Groton, Connecticutt 06340-6096

Appendix A

Introduction

The United States Coast Guard utilizes numerous computer applications in the development and implementation of new Aids to Navigation. These applications range from automatic aid positioning to range design. Even though these applications perform flawlessly, they are unable to communicate the common (or shared) information to one another.

This study was conducted specifically to determine the feasability of integrating the Laptop Automated Aid Positioning System (LAAPS) with the Aids to Navigation Information System (ATONIS Version 2.0).

Background

LAAPS is a tool which was designed to assist in the positioning of Aids to Navigation. It has the capability to automatically capture/collect position information via the serial port of the laptop. The information collected is range, bearing, sextant angles and latitude/longitude information from DGPS. LAAPS was developed using the Pascal programming language and runs on a MS-DOS based laptop computer.

ATONIS is the Coast Guard standard for managing Aids to Navigation and is integral with the Coast Guard's short-range navigation mission. Additionally, ATONIS provides for long term planning, budgeting, forecasting and procurement. It was designed for use on either the USCG Standard Workstation or a MS-DOS based Laptop computer. ATONIS was developed using the Progress 4GL Relational Database Language.

Feasibility Statement

To determine the feasibility of integrating the ATONIS and Laaps software and common data items.

Applied Methods

a. Review and analysis of LAAPS software and data file structures.
b. Review and analysis of ATONIS schema and data types.
c. Determine common data.
d. Development of software to allow sharing of common data.

Results

The review of both LAAPS and ATONIS found only minimal commanality of data. Specifically: aid number, aid name, latitude and longitude, water depth and aid accuracy class.

In LAAPS, aid information is stored in the {location}.AID which is a formatted ascii file where each line is associated with an individual aid. Refer to Appendix A at the end of this document for a full description of the LAAPS aid data file.

In ATONIS, the common aid data is stored in ATONIS database files AID_1 and AID_2. Refer to Appendix B for a full description of the ATONIS AID_1 and AID_2 files.

ATONIS-LAAPS Integration (ALI) software was developed to provide the means for import (from LAAPS to ATONIS) and export (from ATONIS to LAAPS) of the identified command data elements. ALI was developed in the Progress 4GL Relational Database Language to be executed on a MS-DOS based laptop in DOS Protected Mode (DPMI). Refer to Appendix C for the listing of ALI.

Conclusion

Integration of LAAPS and ATONIS data can only be accomplished through the use of custom developed software which takes into account the varied data structures and types and provides the necessary conversion processes. To provide true integration of the LAAPS and ATONIS software applications, both applications must utilitize a common file structure which allows the proper data typing as required by both applications without conversion. ALI provides an example of the conversion processes required unique to ATONIS / LAAPS integration.

With the integration of LAAPS and ATONIS comes the ability to share aids to navigation information, provide consistency and accuracy in the information being tracked, and easy of use for both software applications. Other than the time and personnel needed to further develop a common interface no other cost should be incurred.

Appendix A: LAAPS Aid File Data Dictionary

Field Name	Begin	End	Len	Format	Atonis
Aid Name	1	25	25	X(25)	Yes
Latitude Direction	26	26	1	X	Yes
Latitude Degrees	27	30	4	9999	Yes
Latitude Minutes	31	34	4	9999	Yes
Latitude Seconds	35	41	7	999.999	Yes
Filler	42	42	1	X	
Longitude Direction	43	43	1	X	Yes
Longitude Degrees	44	47	4	9999	Yes
Longitude Minutes	48	51	4	9999	Yes
Longitude Seconds	52	58	7	999.999	Yes
Filler	59	59	1	X	
Accuracy Class	60	60	1	X	Yes
Filler	61	61	1	X	
Tol Rad	62	69	8	99999.99	
Water Depth	70	77	8	99999.99	Yes
Chain Length	78	85	8	99999.99	
Object Tail (1)	86	88	3	999	
Object Tail (2)	89	91	3	999	
Object Tail (3)	92	94	3	999	
Object Tail (4)	95	97	3	999	
Object Tail (5)	98	100	3	999	
Object Tail (6)	101	103	3	999	
Object Tail (7)	104	106	3	999	
Object Tail (8)	107	109	3	999	
Object Tail (9)	110	112	3	999	
Ex-Loran Data	113	208	96	X(96)	
Pair Tail (1,1)	209	210	2	99	
Pair Tail (1,2)	211	212	2	99	
Pair Tail (2,1)	213	214	2	99	
Pair Tail (2,2)	215	216	2	99	
Pair Tail (3,1)	217	218	2	99	
Pair Tail (3,2)	219	220	2	99	
Pair Tail (4,1)	221	222	2	99	
Pair Tail (4,2)	223	224	2	99	
Pair Tail (5,1)	225	226	2	99	
Pair Tail (5,2)	227	228	2	99	
Pair Tail (6,1)	229	230	2	99	
Pair Tail (6,2)	231	232	2	99	
Pair Tail (7,1)	233	234	2	99	
Pair Tail (7,2)	235	236	2	99	
Pair Tail (8,1)	237	238	2	99	
Pair Tail (8,2)	239	240	2	99	
Pair Tail (9,1)	241	242	2	99	
Pair Tail (9,2)	243	244	2	99	
Aid Number	245	250	6	999999	Yes

Appendix B: ATONIS Schema (Aid_1 and Aid_2)

ATONIS (Aid_1) Schema Structure

Field-Name	Data-Type	Format
aid_acclass	char	x(3)
aid_color	char	x(3)
aid_name	char	x(35)
aid_number	inte	ZZ,ZZZZZ9
aid_type	char	x(4)
ap_lat_degrees	inte	Z9
ap_lat_direction	char	x(1)
ap_lat_minutes	inte	Z9
ap_lat_seconds	deci3	Z9.999
ap_long_degrees	inte	ZZ9
ap_long_direction	char	x(1)
ap_long_minutes	inte	Z9
ap_long_seconds	deci3	Z9.999
area_of_work	char	x(6)
base	char	x(3)
battery_backup	char	x(15)
battery_backup_number_of	inte	Z9
battery_main	char	x(15)
battery_main_number_of	inte	Z9
battery_main_replace_interval	inte	Z9
battery_main_replace_last_date	date	99/99/9999
battery_main_replace_next_date	date	99/99/9999
battery_seasonal	char	x(15)
battery_seasonal_number_of	inte	Z9
bottom	char	x(7)
bridle	char	x(8)
chain_addseg_1_diameter	char	x(7)
chain_addseg_1_length	inte	ZZZ9
chain_addseg_1_size	char	x(7)
chain_addseg_2_diameter	char	x(7)
chain_addseg_2_length	inte	ZZZ9
chain_addseg_2_size	char	x(7)
chain_chafe_diameter	char	x(7)
chain_chafe_length	inte	ZZZ9
chain_chafe_size	char	x(7)
chain_riser_diameter	char	x(7)
chain_riser_length	inte	ZZZ9
chain_riser_size	char	x(7)

ATONIS (Aid_2) Schema Structure

Field-Name	Data-Type	Format
aid_number	inte	ZZ,ZZZZZ9
chart_number_01	inte	ZZZZ9
chart_number_02	inte	ZZZZ9
chart_number_03	inte	ZZZZ9
chart_number_04	inte	ZZZZ9
chart_number_05	inte	ZZZZ9
chart_number_06	inte	ZZZZ9
chart_number_07	inte	ZZZZ9
chart_number_08	inte	ZZZZ9
chart_number_09	inte	ZZZZ9
chart_number_10	inte	ZZZZ9
chart_number_11	inte	ZZZZ9
chart_number_12	inte	ZZZZ9
chart_number_13	inte	ZZZZ9
chart_number_14	inte	ZZZZ9
chart_number_15	inte	ZZZZ9
chart_page	char	x(5)
construction_date	date	99/99/9999
daybd_number_of_type_1	inte	9
daybd_number_of_type_2	inte	9
daybd_size_1	char	x(3)
daybd_size_2	char	x(3)
daybd_type_1	char	x(4)
daybd_type_2	char	x(4)
depth	inte	ZZZ9
district_remarks	char[2]	x(65)
district_revision_date	date	99/99/9999
dlc_type_primary	char	x(3)
dlc_type_secondary	char	x(3)
dr_fl	inte	Z9
environment	char	x(3)
fl_type_primary	char	x(3)
fl_type_secondary	char	x(3)
fog_detector	char	x(3)
foundation	char	x(10)
group_jurisdiction	char	x(4)
historical_landmarks	char	x(3)
hull_authorized	char	x(8)
hull_onscene	char	x(8)
hull_onscene_sn	char	x(12)
hull_seasonal	char	x(6)
hull_seasonal_sn	char	x(12)

```
Def Var bl_par_t051 As Char Format "XX" Initial " 0".
Def Var bl_par_t052 As Char Format "XX" Initial " 0".
Def Var bl_par_t061 As Char Format "XX" Initial " 0".
Def Var bl_par_t062 As Char Format "XX" Initial " 0".
Def Var bl_par_t071 As Char Format "XX" Initial " 0".
Def Var bl_par_t072 As Char Format "XX" Initial " 0".
Def Var bl_par_t081 As Char Format "XX" Initial " 0".
Def Var bl_par_t082 As Char Format "XX" Initial " 0".
Def Var bl_par_t091 As Char Format "XX" Initial " 0".
Def Var bl_par_t092 As Char Format "XX" Initial " 0".
Def Var bl_aid_nmbr As Char Format "XXXXXX".

Def Var vl_aid_nmbr Like atonis.aid_1.aid_number.
Def Var vl_aid_name Like atonis.aid_1.aid_name.
Def Var vl_lat_dir Like atonis.aid_1.ap_lat_direction.
Def Var vl_lat_deg Like atonis.aid_1.ap_lat_degrees.
Def Var vl_lat_min Like atonis.aid_1.ap_lat_minutes.
Def Var vl_lat_sec Like atonis.aid_1.ap_lat_seconds.
Def Var vl_lng_dir Like atonis.aid_1.ap_long_direction.
Def Var vl_lng_deg Like atonis.aid_1.ap_long_degrees.
Def Var vl_lng_min Like atonis.aid_1.ap_long_minutes.
Def Var vl_lng_sec Like atonis.aid_1.ap_long_seconds.
Def Var vl_acc_clas Like atonis.aid_1.aid_acclass.
Def Var vl_wtr_dep Like atonis.aid_2.depth.

/* ---------------------------------------------------------------------- */
/*                     Form Statements                     */
/* ---------------------------------------------------------------------- */

Form Skip(1)
  "    I - Import from LAAPS to ATONIS         " Skip
  "    E - Export from ATONIS to LAAPS         "
  Skip(1)
  "     Option (F4/Cancel to Quit)?" AL_Option
  Skip(1)
  with Frame Laaps_Atonis Centered Overlay No-Labels Row 7
  Title " [ LAAPS - ATONIS Interface ] ".

Form Skip(1)
  "    Area-of-Work to Export (F4/Cancel to Abort)?" AOWork "         "
  Skip(1)
  with Frame Exp_aowork Centered Overlay No-Labels Row 12.

Form Skip(1)
  "    Enter the full Filename specification of the LAAPS AID        " Skip
  "    data file to Import from or Export to.                 "
  Skip(1)
  "      " ImEx_File
  Skip(1)
  with Frame IE_getfilename Centered Overlay No-Labels Row 12.

Form Skip(1)
  "  ___Aid #___  _________________________________________  " Skip
  vl_aid_nmbr at 6 vl_aid_name at 18                  Skip(1)
  with frame vl_disply_aid overlay no-labels centered row 13
     title " [ Aid Import Status ] ".

Form Skip(1)
  "  # of Aids Imported from LAAPS:"
  Laap_Count " of " Laap_Total "    "          Skip(1)
  "          Press F1 to Continue"  kk        Skip(1)
  With Frame aid_in_count Overlay No-Labels Centered Row 10
     Title " [ Aid Import Count ] ".
```

Appendix C: Atonis/Laaps Interface (ALI) Listing

```
/* ================================================================ */
/* Atonis / Laaps Interface (ALI)*/
/* ================================================================ */

/* ---------------------------------------------------------------- */
/* Procedure Name:      ALI.P */
/* Author:              Melvin C WIley (UNISYS) */
/* ---------------------------------------------------------------- */

/* ---------------------------------------------------------------- */
/*                 Variable Definitions                 */
/* ---------------------------------------------------------------- */

Def Var AOWork like atonis.aid_1.area_of_work.

Def Var j as Integer.
Def Var kk as Character Format "X".
Def Var Laap_Count as Integer format "999".
Def Var Laap_Total as Integer format "999".
Def Var AL_Option as Character Format "!".
Def Var ImEx_File as Character Format "X(48)".
Def Var Imp_Error as Logical.
Def Var Imp_Found as Logical.
Def Var myline as Character Format "X(255)".
Def Var doaid_create as Character Format "!".

Def Var bl_aid_name As Char Format "X(25)".
Def Var bl_lat_dir As Char Format "X".
Def Var bl_lat_deg As Char Format "XXXX".
Def Var bl_lat_min As Char Format "XXXX".
Def Var bl_lat_sec As Char Format "XXXXXXX".
Def Var bl_fillr_1 As Char Format "X" initial " ".
Def Var bl_lng_dir As Char Format "X".
Def Var bl_lng_deg As Char Format "XXXX".
Def Var bl_lng_min As Char Format "XXXX".
Def Var bl_lng_sec As Char Format "XXXXXXX".
Def Var bl_fillr_2 As Char Format "X" Initial " ".
Def Var bl_acc_clas As Char Format "X".
Def Var bl_fillr_3 As Char Format "X" Initial " ".
Def Var bl_tol_rad As Char Format "XXXXXXXX" Initial "        0.00".
Def Var bl_wtr_dep As Char Format "XXXXXXXX".
Def Var bl_chn_len As Char Format "XXXXXXXX" Initial "        0.00".
Def Var bl_obj_t001 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t002 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t003 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t004 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t005 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t006 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t007 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t008 As Char Format "XXX" Initial "  0".
Def Var bl_obj_t009 As Char Format "XXX" Initial "  0".
Def Var bl_ex_loran As Char Format "X(96)".
Def Var bl_par_t011 As Char Format "XX" Initial " 0".
Def Var bl_par_t012 As Char Format "XX" Initial " 0".
Def Var bl_par_t021 As Char Format "XX" Initial " 0".
Def Var bl_par_t022 As Char Format "XX" Initial " 0".
Def Var bl_par_t031 As Char Format "XX" Initial " 0".
Def Var bl_par_t032 As Char Format "XX" Initial " 0".
Def Var bl_par_t041 As Char Format "XX" Initial " 0".
Def Var bl_par_t042 As Char Format "XX" Initial " 0".
```

```
/* ------------------------------------------------------------------------ */
/*                  Main Procedure Code                    */
/* ------------------------------------------------------------------------ */

Do_Imp_Exp:
Repeat:

  AL_option = "".
  View Frame Laaps_Atonis.
  Update AL_option Auto-Return Go-On( F1 F4 CANCEL ESCAPE RETURN )
    with Frame Laaps_Atonis.
    If Lastkey = Keycode( "F4") or Lastkey = Keycode( "CANCEL") or
      Lastkey = Keycode( "ESCAPE") or Trim(AL_option) = "" then Do:
    Hide Frame Laaps_Atonis No-Pause.
    Leave Do_Imp_Exp.
  End.
  If Index("IE", AL_option) = 0 then Next Do_Imp_Exp.

  Do_AOW:
  Repeat:

    If AL_option = "E" Then Do:

      AOWork = "".
      View Frame Exp_aowork.

      Update AOWork Go-On( F1 F4 CANCEL ESCAPE RETURN )
      with Frame Exp_aowork.

      If Lastkey = Keycode( "F4") or Lastkey = Keycode( "CANCEL") or
        Lastkey = Keycode( "ESCAPE") or Trim(AOWork) = "" then Do:
        Hide Frame Exp_aowork No-Pause.
        Next Do_Imp_Exp.
      End.

    End. /* If AL_option = "E" */

    Do_IMEX:
    Repeat:

      ImEx_File = "".
      View Frame IE_getfilename.

      Update ImEx_File
           Go-On( F1 F4 CANCEL ESCAPE RETURN )
           with Frame IE_getfilename.

      If Lastkey = Keycode( "F4") or Lastkey = Keycode( "CANCEL") or
        Lastkey = Keycode( "ESCAPE") or Trim(ImEx_File) = "" then Do:
        Hide Frame IE_getfilename No-Pause.
        If AL_option = "I" then Next Do_Imp_Exp.
        If AL_option = "E" then Next Do_AOW.
      End.

      Leave Do_AOW.

    End. /* Repeat - Do_IMEX */

  End. /* Repeat - Do_AOW */

  If AL_option = "E" then Do:

    Output to value(ImEx_File).
```

```
bl_ex_loran = fill("x",96). j = 1.

first_fifty:
for each aid_1 where AOWork = area_of_work,
   each aid_2 of aid_1:
   if j > 50 then leave first_fifty.
   bl_aid_name = substr(trim(aid_1.aid_name) + fill(" ",35),1,25).
   bl_lat_dir  = trim(  aid_1.ap_lat_direction).
   bl_lat_deg  = string(aid_1.ap_lat_degrees,"ZZZ9").
   bl_lat_min  = string(aid_1.ap_lat_minutes,"ZZZ9").
   bl_lat_sec  = string(aid_1.ap_lat_seconds,"ZZ9.999").
   bl_lng_dir  = trim(  aid_1.ap_long_direction).
   bl_lng_deg  = string(aid_1.ap_long_degrees,"ZZZ9").
   bl_lng_min  = string(aid_1.ap_long_minutes,"ZZZ9").
   bl_lng_sec  = string(aid_1.ap_long_seconds,"ZZ9.999").
   bl_acc_clas = substring(trim(aid_1.aid_acclass),1,1).
   bl_wtr_dep  = string(aid_2.depth,"ZZZZ9.99").
   bl_aid_nmbr = string(aid_1.aid_number,"ZZZZZ9").
   Pause 0.
   Display bl_aid_name at   1 bl_lat_dir  at  26 bl_lat_deg  at  27
         bl_lat_min  at  31 bl_lat_sec  at  35 bl_fillr_1  at  42
         bl_lng_dir  at  43 bl_lng_deg  at  44 bl_lng_min  at  48
         bl_lng_sec  at  52 bl_fillr_2  at  59 bl_acc_clas at  60
         bl_fillr_3  at  61 bl_tol_rad  at  62 bl_wtr_dep  at  70
         bl_chn_len  at  78 bl_obj_t001 at  86 bl_obj_t002 at  89
         bl_obj_t003 at  92 bl_obj_t004 at  95 bl_obj_t005 at  98
         bl_obj_t006 at 101 bl_obj_t007 at 104 bl_obj_t008 at 107
         bl_obj_t009 at 110 bl_ex_loran at 113 bl_par_t011 at 209
         bl_par_t012 at 211 bl_par_t021 at 213 bl_par_t022 at 215
         bl_par_t031 at 217 bl_par_t032 at 219 bl_par_t041 at 221
         bl_par_t042 at 223 bl_par_t051 at 225 bl_par_t052 at 227
         bl_par_t061 at 229 bl_par_t062 at 231 bl_par_t071 at 233
         bl_par_t072 at 235 bl_par_t081 at 237 bl_par_t082 at 239
         bl_par_t091 at 241 bl_par_t092 at 243 bl_aid_nmbr at 245
   Skip
   with Frame Sample_line No-Attr-Space No-Labels No-Box Width 255.

  j = j + 1.

 End. /* for */

 Output to Terminal.
 Hide Frame IE_getfilename No-Pause.
 Hide Frame Exp_aowork No-Pause.

End. /* If AL_option = "E" (export from ATONIS) */


If AL_option = "I" then Do:

 DOS silent quoter value(ImEx_File) >LAAPATON.IMP.
 Input From Value(Search("LAAPATON.IMP")) No-Echo.

 View Frame vl_disply_aid.

 Laap_Count = 0.
 Laap_Total = 0.

 Laapin:
 Repeat:

  Import myline.
```

```
myline = substring(trim(myline) + fill(" ",250),1,250).

vl_aid_nmbr = integer(substring(myline,245,6)).
vl_aid_name = substring(myline,1,25).
vl_lat_dir  = substring(myline,26,1).
vl_lat_deg  = integer(substring(myline,27,4)).
vl_lat_min  = integer(substring(myline,31,4)).
vl_lat_sec  = decimal(substring(myline,35,7)).
vl_lng_dir  = substring(myline,43,1).
vl_lng_deg  = integer(substring(myline,44,4)).
vl_lng_min  = integer(substring(myline,48,4)).
vl_lng_sec  = decimal(substring(myline,52,7)).
vl_acc_clas = substring(myline,60,1).
vl_wtr_dep  = integer(substring(myline,70,8)).

Imp_Found = no.
Imp_Error = no.

If (vl_aid_nmbr = ?) or (vl_aid_nmbr < 1) then Imp_Error = yes.
Imp_Found = Can-find(atonis.aid_1
        Where atonis.aid_1.aid_number = vl_aid_nmbr).

Laap_Total = Laap_Total + 1.

If (Not Imp_Found) or Imp_Error then Next Laapin.

Laap_Count = Laap_Count + 1.

If length(trim(vl_aid_name)) < 1
  then vl_aid_name = "**** Undefined Aid Name ***".

If vl_lat_dir  = ? or (vl_lat_dir <> "N" and vl_lat_dir <> "S")
  then vl_lat_dir = "N".

If vl_lng_dir  = ? or (vl_lng_Dir <> "E" and vl_lng_dir <> "W")
  then vl_lng_dir = "W".

Find Atonis.Aid_1 Where atonis.aid_1.aid_number = vl_aid_nmbr
    No-Error No-Wait.

If Available Atonis.Aid_1 then Do:
  aid_1.ap_lat_direction  = vl_lat_dir.
  aid_1.ap_lat_degrees    = vl_lat_deg.
  aid_1.ap_lat_minutes    = vl_lat_min.
  aid_1.ap_lat_seconds    = vl_lat_sec.
  aid_1.ap_long_direction = vl_lng_dir.
  aid_1.ap_long_degrees   = vl_lng_deg.
  aid_1.ap_long_minutes   = vl_lng_min.
  aid_1.ap_long_seconds   = vl_lng_sec.
End.

Find Atonis.Aid_1 Where atonis.aid_1.aid_number = vl_aid_nmbr
    No-Error No-Wait.

If Available Atonis.Aid_2 then aid_2.depth = vl_wtr_dep.

Display vl_aid_nmbr vl_aid_name with frame vl_disply_aid.

End. /* Repeat */

Input Close.

Hide Frame vl_disply_aid no-pause.
Hide Frame IE_getfilename no-pause.
```

```
    View Frame aid_in_count.
    Display Laap_Count Laap_Total With Frame aid_in_count.

    Update kk Go-On( F1 F4 ESCAPE CANCEL RETURN TAB )
        With Frame aid_in_count.

    Hide Frame aid_in_count.

  End. /* If AL_option = "I" (import from LAAPS) */

End. /* Repeat Do_Imp_Exp */

Return.
```

United States Coast Guard
Research and Development Center

Avery Point, Groton, Connecticut 06340-6096

Atonis - Solar Calc
Integration Feasibility Study

Prepared By:

Melvin C Wiley
Unisys Corporation

Introduction

The United States Coast Guard uses a significant number of software applications in the development, implementation and management of new and existing Aids-to-Navigation. Even though the majority of these software applications perform without flaw, they are still unable to share the information which is common between one another.

This Study focuses on the integration of Solar Calc program, revision 4.0, and the Aids-to-Navigation Information System (ATONIS) utilizing the Progress 4GL programming language.

Background

Solar Calc is the tool currently employed by the Coast Guard to determine the size of a solar panel array and the number of solar batteries required when solarizing a navigational aid. The Solar Calc program was developed in the Basic programming language and compiled to run on the USCG Standard Workstation.

Atonis provides standard procedures by which the Coast Guard manages short range aids-to-navigation and is integral with the Coast Guard's short-range navigation mission. With Atonis, the Coast Guard has the necessary information readily available for development of long and short range logistic and operational plans. Atonis was developed in the Progress 4GL programming language.

Feasibility Statement

Integration of the Solar Calc program/function into ATONIS using the Progress 4GL programming language.

Applied Methods

- Review and analysis of Solar Calc software and data files.
- Review of the ATONIS database elements.
- Determination of common data.
- Development of a Progress 4GL software application compatable with ATONIS data.

Analysis Results

Analysis of both Solar Calc and ATONIS data file elements revealed that aid position latitude was the only data element common between the two applications.

Feasibility Conclusion

Integration of the Solar Calc and ATONIS software applications can be accomplished with the development of a Progress 4GL application capable of reading the necessary information contained in the Solar Calc and ATONIS data files, applying the mathematical methods contained in the Solar Calc application to the data and displaying the results.

Atonis Solar Calc (ASC) Screen Layouts

```
United States Coast Guard                              Atonis Solar Calc v5.0

                        -=* Specifications *=-

  Aid Number.: _____                                          Latitude.: __
        Name.: ___________________________                   Longitude.: __

  Date of Installation.: __/__/___               Duration of Analysis.: __

  Solar Panel Max Power Point.: __.__     Solar Battery Size.......: __
  Solar Panel Size (in Watts).: __        Initial Battery Capacity.: __

  Array Tilt Angle (degs).....: __
                                           ( F4/CANCEL/ESCAPE to Abort )
  F6: Electrical Loads Screen              ( F1 to obtain results      )

 You must complete both the Specifications and Electrical Loads screens
 before you can obtain valid results.  To obtain results, Press F1.
```

Figure 1 - ASC Specifications Screen

```
United States Coast Guard                              Atonis Solar Calc v5.0

                        -=* Electrical Loads *=-

  Number of Electrical Loads: __ ( maximum of 10 electrical loads )

   Load [??] -> Average Current Drain    : _.__ ( in amps            )
                Duty Cycle               : __   ( in percent         )
                Daily Hours of Operation : __   ( 0 = NightTime Only )
                CG-181 Use Required      : _    ( Y = Yes, N = No    )


                                           ( F4/CANCEL/ESCAPE to Abort )
  F5: Specifications Screen                ( F1 to obtain results      )

 You must complete both the Specifications and Electrical Loads screens
 before you can obtain valid results.  To obtain results, Press F1.
```

Figure 2 - ASC Electrical Loads Screen

```
United States Coast Guard                              Atonis Solar Calc v5.0

                            -=* Results *=-

  < >Array Size...........: ????.??    Load Amp-Hrs (Min).: ????.?? AH
  < >Battery Size.........: ????.??                (Avg).: ????.?? AH
  { }Min Battery Capacity.:   ??.??                (Max).: ????.?? AH

  Solar Panel Qty (10/20/35): ??/??/??      Solar Bat Qty (100AH).: ??

  Notes:  <*> = indicates a non-standard solar panel / battery size.
          {-} = indicates that minimum battery capacity is below 48%.
          {+} = indicates that minimum battery capacity is above 52%.

                      ( F4/CANCEL/ESCAPE to Abort )

  F5: Specifications Screen   F6: Electrical Loads Screen    F7: Print
```

Figure 3 - ASC Results Screen

Recommended Progress Database Schema for ASC

Filename: AidInfo

Order	Field-Name	dType	Ext	Flgs	Format	Initial
1000	Aid_Number	inte		I	ZZ,ZZZZZ9	0
1010	Aid_Pos_Latitude	inte			99	0
1020	Aid_Pos_Longitude	inte			999	0
1030	Duration_of_Analysis	deci0			99	5
1040	EL_Avg_Crnt_Drain	deci2	10		99.99	0.25
1050	EL_CG181_Use	logi	10		yes/no	yes
1060	EL_Duty_Cycle	deci0	10		99	10
1070	EL_Num_Loads	inte			99	1
1080	EL_Op_Hours	deci0	10		99	0
1090	Install_Date	date			99/99/9999	?
1100	Max_Pwr_Pnt	deci1			99.9	13.8
1110	Rslts_Array_Size	deci0			999	0
1120	Rslts_Battery_Capacity	deci2			99.99	0
1130	Rslts_Battery_Size	deci0			9999	0
1140	Rslts_Load_Amps	deci2	3		999.99	0
1150	Rslts_Sobat_Qty	inte			99	0
1160	Rslts_Sopan_Qty	inte	3		99	0
1170	Solar_Array_Size	deci0			999	10
1180	Solar_Array_Tilt	deci0			99	60
1190	Solar_Battery_Capacity	deci0			9999	100
1200	Solar_Battery_Size	deci0			9999	100
1210	Std_Insol_Dat_Index	inte			999	1
1220	User_Insol_Data_Array_1	deci2	12		99999.99	0
1230	User_Insol_Data_Array_2	deci2	12		99999.99	0
1240	Tmatrix_Index	inte			999	0

+ Data Dictionary Report Legend +

c - field is case-sensitive　　m - field is mandatory
i - field participates in an index　　v - field is a view component

Index Name (* indicates primary)	Unique	Field Name	Seq	Asc	Abbr
P_Index*	yes	Aid_Number	1	yes	no

Filename: SolarData

Order	Field-Name	dType	Ext	Flgs	Format	Initial
1000	Std_Insol_Dat_Index	inte		I	999	0
1010	Location_City	char		I	x(16)	?
1020	Location_State	char		I	XX	?
1030	Insol_Data_Array_1	deci2	12		99999.99	0
1040	Insol_Data_Array_2	deci2	12		99999.99	0
1050	Latitude_Mn	inte		I	99	0
1060	Latitude_Mx	inte		I	99	0

+ Data Dictionary Report Legend +

c - field is case-sensitive m - field is mandatory
i - field participates in an index v - field is a view component

Index Name (* indicates primary)	Unique	Field Name	Seq	Asc	Abbr
P_Index*	yes	Std_Insol_Dat_Index	1	yes	no
S_Index	yes	Latitude_Mn	1	yes	no
		Latitude_Mx	2	yes	no
T_Index	no	Location_City	1	yes	no
		Location_State	2	yes	no

Filename: Tmatrix

Order	Field-Name	dType	Ext	Flgs	Format	Initial
1000	Tmatrtix_Index	inte		I	999	0
1010	Tmatrix_Info	deci2	24		99999.99	0

+ Data Dictionary Report Legend +

c - field is case-sensitive m - field is mandatory
i - field participates in an index v - field is a view component

Index Name (* indicates primary)	Unique	Field Name	Seq	Asc	Abbr
P_Index*	yes	Tmatrix_Index	1	yes	no

SolrData.Dat

Data contained in the SolrData data file is used in computing the total average number of hours of sunlight in preparation for the Solar Array and Battery Size calculations. If is a text file and can be edited with any ascii text editor. The structure of the SolrData file is as follows:

City
State
Latitude
12 Number Array (Insolation values associated with the 1st half of each month)
12 Number Array (Insolation values associated with the 2nd half of each month)

The following listing is dump of the data contained in the SolrData file.

ADAK, AK											
52.00											
02.62	04.91	08.13	11.72	13.39	13.42	12.72	10.77	08.62	06.00	03.50	02.13
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
ALPENA, MI											
45.10											
04.11	07.00	11.67	15.97	19.52	21.32	21.39	17.97	13.12	08.42	04.34	03.07
26.90	28.70	36.80	51.60	63.60	74.30	79.00	77.30	68.40	58.90	42.90	31.00
ANNETTE, AK											
55.00											
02.02	04.25	08.14	13.05	16.72	16.63	16.33	13.19	09.21	04.79	02.48	01.39
38.00	41.50	43.70	48.80	56.20	61.00	64.00	64.60	59.80	51.70	44.30	40.10
APALACHICOLA, FL											
29.70											
09.67	12.78	16.73	21.32	23.73	22.68	20.58	19.15	17.43	15.56	11.80	09.28
61.10	63.10	67.60	75.00	81.80	86.40	87.50	87.70	84.70	78.30	69.00	62.70
ARCATA, CA											
41.00											
06.00	09.00	12.86	18.01	20.91	22.26	20.52	17.92	15.23	10.62	06.73	05.33
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
ASTORIA, OR											
46.10											
03.57	06.19	09.83	14.22	18.25	18.45	19.82	17.01	13.43	08.09	04.40	02.96
46.50	50.60	52.10	55.60	60.30	63.80	67.70	68.30	67.60	61.00	53.40	48.60
BALTIMORE, MD											
39.20											
06.66	09.53	13.19	16.89	19.45	21.33	20.69	18.15	15.10	11.32	07.49	05.67
41.90	43.90	53.00	65.20	74.80	83.20	86.70	85.10	79.00	68.30	56.10	43.90
BARBERS POINT, HI											
21.30											
13.71	16.35	18.67	20.81	22.39	22.98	22.91	22.37	20.60	17.65	14.75	13.23
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
BARROW, AK											
71.30											
00.00	00.84	05.57	11.92	12.94	17.34	16.56	09.71	04.70	01.43	00.04	00.00
-08.00	-12.60	-08.60	06.50	24.20	37.10	44.30	42.00	33.40	20.20	05.40	-06.40

BETHEL, AK											
60.80											
01.10	03.59	08.38	13.62	16.49	17.23	14.64	10.44	07.95	04.20	01.53	00.55
12.30	15.80	20.30	32.70	48.50	60.20	61.90	58.50	51.80	36.10	23.60	11.30
BOSTON, MA											
42.40											
05.40	08.05	11.54	15.05	18.39	20.62	19.85	16.87	14.30	10.10	05.71	04.57
35.90	37.50	44.60	56.30	67.10	76.60	81.40	79.30	72.20	63.20	51.70	39.30
BROWNSVILLE, TX											
25.90											
10.36	12.89	16.55	19.72	21.87	24.01	25.11	23.01	19.22	16.33	11.97	09.79
69.50	72.70	76.60	83.10	87.10	90.60	92.80	93.00	89.90	84.70	77.50	72.30
BUFFALO, NY											
42.90											
03.96	06.20	10.08	14.92	18.12	20.47	20.16	17.17	13.07	08.89	04.58	03.22
29.80	31.00	39.00	53.30	64.30	75.10	79.50	77.60	70.80	60.20	46.10	33.60
CAPE HATTERAS, NC											
35.30											
07.78	10.81	15.05	20.13	22.26	23.11	21.80	19.36	16.69	12.90	09.91	07.48
52.30	53.10	57.90	66.30	73.80	80.50	83.80	83.40	79.50	71.30	63.10	54.80
CHARLESTON, SC											
32.90											
08.45	11.30	15.19	19.66	21.11	20.93	20.42	17.99	15.82	13.54	10.60	08.18
59.80	61.90	67.80	76.20	83.10	87.70	89.10	88.60	84.50	77.10	68.40	60.80
CHATTANOOGA, TN											
35.00											
07.16	09.74	13.35	17.59	19.66	20.78	19.69	18.50	15.16	12.58	08.76	06.59
49.90	53.40	61.20	72.90	81.00	87.50	89.50	89.00	83.40	73.50	60.70	50.90
CHERRY POINT, NC											
34.90											
08.59	11.64	15.74	20.36	21.84	22.00	20.77	18.55	16.20	13.27	10.29	08.15
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
CHICAGO, IL											
41.80											
05.75	08.62	12.56	16.56	20.30	22.78	22.06	19.51	15.37	11.00	06.42	04.56
31.50	34.60	44.60	59.30	70.30	80.60	84.40	83.30	75.80	65.00	48.10	35.30
CINCINNATI, OH											
39.10											
05.68	08.38	11.66	15.87	18.98	20.85	20.10	18.55	14.89	11.23	06.68	04.91
39.70	42.70	51.80	65.00	74.40	83.20	86.50	85.80	79.70	68.50	53.20	42.00
CLEVELAND, OH											
41.40											
04.41	06.82	10.47	15.32	19.08	20.92	20.75	17.96	14.07	09.84	05.29	03.61
33.40	35.00	44.10	58.00	68.40	78.20	81.60	80.40	74.20	63.60	48.80	36.40
CORPUS CHRISTI, TX											
27.80											
10.19	13.02	16.23	18.64	21.18	23.76	24.81	22.59	19.15	16.07	11.83	09.59
66.50	69.80	75.50	82.10	86.60	91.20	94.40	94.80	90.00	84.10	75.20	69.30
DAYTONA BEACH, FL											
29.20											
10.88	13.77	17.57	21.38	22.34	20.72	20.25	19.09	16.77	14.20	11.75	09.88
69.10	70.40	74.50	80.20	85.20	88.40	89.60	89.40	87.00	81.40	75.10	70.20

DETROIT, MI
42.40

04.74	07.72	11.35	15.88	19.47	21.18	20.83	17.88	14.22	09.93	05.42	03.90
31.70	33.70	43.10	57.60	68.50	79.10	83.10	81.60	74.20	63.40	47.70	35.40

DULUTH, MN
46.80

04.41	07.64	11.74	15.58	18.64	20.06	21.04	17.56	12.43	08.23	04.32	03.31
17.60	22.10	32.60	47.80	60.00	69.70	76.40	74.40	64.00	54.30	35.30	22.50

FORT SMITH, AR
35.30

08.43	11.34	14.89	18.34	21.70	23.71	23.44	21.31	17.04	13.63	09.66	07.74
49.90	54.60	62.10	74.20	81.30	89.00	93.80	93.50	86.30	76.50	62.70	52.20

GREEN BAY, WI
44.50

05.12	08.23	12.53	16.33	19.51	21.65	21.43	18.41	13.82	09.30	05.28	03.97
23.90	27.20	37.10	54.10	65.80	75.80	80.70	79.10	69.80	59.60	41.80	28.60

GUANTANAMO BAY, CU
19.90

15.92	18.70	21.86	24.06	23.13	22.25	23.63	22.73	20.70	17.98	16.25	14.90
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

HILO, HI
19.70

12.71	14.14	15.31	16.28	17.63	18.82	18.44	18.07	17.55	15.57	12.54	11.57
79.60	79.40	78.80	79.80	81.30	82.70	83.00	83.50	83.60	83.20	81.30	79.40

HOMER, AK
59.60

01.38	03.79	08.62	14.17	17.96	19.87	18.14	13.49	08.97	04.96	01.99	00.73
28.00	31.80	35.00	42.30	50.30	56.70	60.10	60.10	54.80	44.40	34.50	27.60

HONOLULU, HI
21.30

13.39	15.85	18.40	20.38	22.12	22.75	22.72	22.32	20.54	17.48	14.37	12.85
79.30	79.20	79.70	81.40	83.60	85.60	86.80	87.40	87.40	85.80	83.20	80.30

HOUGHTON, MI
47.20

02.77	05.49	10.59	15.50	18.84	20.86	20.86	17.27	11.46	07.61	03.30	02.18
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

JACKSONVILLE, FL
30.50

10.21	13.21	17.27	21.06	22.20	21.40	20.45	19.23	16.37	13.88	11.30	09.28
64.60	66.90	72.20	79.00	84.60	88.30	90.00	89.70	86.00	79.20	71.40	65.60

JUNEAU, AK
58.40

01.32	03.21	06.92	11.87	14.66	16.05	14.51	11.17	07.25	03.64	01.69	00.70
29.10	33.90	38.20	46.50	55.40	62.00	63.60	62.30	56.10	47.20	37.30	32.00

KANSAS CITY, MO
39.30

07.35	10.15	13.65	17.88	21.25	23.60	23.86	21.14	16.48	12.40	08.37	06.37
35.70	41.40	50.70	64.70	74.20	82.80	88.00	87.20	78.80	68.20	51.40	39.30

KING SALMON, AK
58.70

01.66	04.28	09.07	13.68	16.82	17.48	15.70	11.86	08.83	05.38	02.31	01.03
21.00	24.60	28.60	39.10	51.30	59.70	62.50	60.90	54.80	40.80	29.00	19.60

KWAJALEIN ISLAND, PI

08.70

17.86	19.88	20.39	19.75	18.46	18.28	18.27	19.15	18.33	17.31	16.55	16.46
85.00	85.40	85.90	85.80	86.00	86.20	86.70	87.30	87.30	87.20	86.00	85.50

KODIAK, AK

57.60

01.69	04.04	08.87	13.71	15.62	17.36	15.98	13.21	09.01	05.55	02.34	01.10
34.50	35.70	36.90	41.60	47.90	54.60	59.10	60.10	54.90	45.60	39.00	34.30

KOROR ISLAND, PI

07.30

15.85	17.69	18.51	19.20	17.79	16.52	16.37	16.83	17.33	17.10	16.44	15.19
86.40	86.50	87.10	88.10	88.00	87.70	87.10	87.00	87.50	87.90	88.00	87.20

LEWISTON, ID

46.40

03.86	06.91	11.57	16.29	20.91	22.87	26.51	21.92	16.28	09.76	04.69	03.25
37.90	46.00	52.90	62.00	70.60	77.90	89.20	87.10	77.60	63.20	48.00	41.00

LIHUE, HI

22.00

12.52	14.75	16.75	18.62	20.70	21.20	21.14	20.63	19.77	16.45	13.10	11.95
77.90	77.90	77.90	79.20	81.40	83.30	84.00	84.60	84.80	83.30	80.80	78.20

LITTLE ROCK, AR

34.70

08.30	11.38	14.90	18.28	21.90	23.91	23.06	21.12	17.23	13.94	09.62	07.65
50.10	53.80	61.80	73.50	81.40	89.30	92.60	92.60	85.80	76.00	62.40	52.10

LONG BEACH, CA

33.80

10.53	13.79	18.27	21.99	23.43	24.29	26.10	23.83	19.30	15.05	11.39	09.60
65.20	66.20	67.70	70.50	73.80	76.80	82.80	84.00	83.30	78.30	72.70	66.90

LOS ANGELES, CA

33.90

10.51	13.78	18.37	22.14	23.37	24.05	26.19	23.60	19.08	14.95	11.39	09.63
63.50	64.10	64.30	65.90	68.40	70.30	74.80	75.80	75.70	72.90	69.60	66.50

LOUISVILLE, KY

38.20

06.19	08.96	12.51	16.65	19.52	21.60	20.85	19.07	15.45	11.83	07.41	05.54
42.00	45.00	54.00	66.90	75.60	83.70	87.30	86.80	80.50	70.30	54.90	44.10

MASSENA, NY

44.90

04.44	07.04	11.09	15.24	18.31	20.19	19.87	16.84	12.75	08.36	04.40	03.34
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

MEMPHIS, TN

35.00

07.75	10.72	14.51	18.60	21.39	23.20	22.38	20.70	16.69	13.67	09.26	07.13
49.40	53.10	60.80	72.70	81.20	88.70	91.60	90.60	84.30	74.90	61.50	51.70

MIAMI, FL

25.80

12.00	14.91	18.20	21.10	20.92	19.38	20.01	18.50	16.53	14.78	12.70	11.57
75.60	76.60	79.50	82.70	85.30	88.00	89.10	89.90	88.30	84.60	79.90	76.60

MILWAUKEE, WI

42.90

05.44	08.36	12.36	16.37	20.07	22.44	22.26	19.51	14.87	10.30	05.95	04.30
27.30	30.30	39.40	54.60	65.00	75.30	80.40	79.70	71.50	61.40	44.40	31.50

MINNEAPOLIS-ST. PAUL, MN
44.90

05.27	08.67	12.52	16.36	19.72	21.88	22.36	19.15	14.24	09.76	05.45	04.01
21.20	25.90	36.90	55.50	67.90	77.10	82.40	80.80	70.70	60.70	40.60	26.60

MOBILE, AL
30.70

09.39	12.48	15.97	19.54	21.25	21.21	19.47	18.63	16.45	14.74	10.84	08.62
61.10	64.10	69.50	78.00	85.00	89.80	90.50	90.60	86.50	79.70	69.50	63.00

NEW ORLEANS, LA
30.00

09.46	12.62	16.06	20.20	22.33	22.74	20.58	19.48	17.18	15.15	11.04	08.84
62.30	65.10	70.40	78.40	84.90	89.60	90.40	90.60	86.60	79.90	70.30	64.20

CENTRAL PARK, NY
40.80

05.68	08.18	11.77	15.48	18.57	19.41	19.16	16.83	13.77	10.16	06.05	04.59
38.50	40.20	48.40	60.70	71.40	80.50	85.20	83.40	76.80	66.80	54.00	41.40

LA GUARDIA, NY
40.80

06.22	09.01	12.68	16.53	19.18	20.45	20.25	17.97	14.53	10.79	06.73	05.18
37.70	39.20	47.10	59.30	69.80	79.40	84.10	82.10	75.20	65.10	53.20	41.00

NOME, AK
64.50

00.34	02.54	07.16	13.46	17.85	19.90	16.05	11.27	07.64	03.47	00.74	00.03
13.50	13.70	16.60	27.00	41.40	52.20	55.80	54.60	48.20	34.40	22.10	11.70

NORFOLK, VA
36.90

07.70	10.58	14.54	19.03	21.42	22.70	21.03	19.07	15.84	12.29	09.21	07.08
48.80	50.00	57.30	67.70	76.20	83.50	86.60	84.90	79.60	70.10	60.50	50.60

NORTH BEND, OR
43.40

04.98	08.00	12.01	17.14	21.08	22.63	23.92	20.27	15.63	10.13	05.95	04.32
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

NORTH OMAHA, NE
41.40

07.20	10.12	13.87	17.69	21.25	24.09	23.91	21.09	15.58	11.91	07.31	05.80
29.10	34.80	44.10	61.00	71.40	80.20	85.40	84.00	75.20	65.90	47.40	34.30

OAKLAND, CA
37.70

08.03	11.55	16.53	21.81	25.10	26.67	26.36	23.30	19.31	13.76	09.33	07.34
54.50	58.00	60.20	62.80	65.40	68.50	69.70	70.20	72.30	68.70	62.00	55.50

PATUXENT RIVER, MD
38.30

06.90	09.78	13.41	17.46	20.01	21.48	20.62	18.47	15.40	11.59	08.02	06.09
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

PHILADELPHIA, PA
39.90

06.30	09.01	12.58	16.27	18.84	20.56	19.95	17.87	14.54	10.88	07.03	05.34
40.10	42.20	51.20	63.50	74.10	83.00	86.80	84.80	78.40	67.90	55.50	43.20

POINT MUGU, CA
34.10

10.52	13.85	18.56	22.14	22.90	23.32	24.04	21.96	18.25	14.71	11.42	09.71
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

PORT ARTHUR, TX											
29.90											
09.08	12.15	15.36	18.27	21.23	22.82	20.95	19.71	17.33	15.00	10.81	08.55
61.50	65.00	70.50	78.30	84.30	89.90	92.00	92.60	88.60	81.30	70.90	64.20
PORTLAND, ME											
43.60											
05.11	07.74	11.00	14.80	17.79	19.43	18.83	16.58	13.14	09.33	05.21	04.12
31.20	33.30	40.80	52.80	63.60	73.20	79.10	77.60	69.90	60.20	47.50	34.90
ROCHESTER, NY											
43.10											
04.13	06.35	10.25	15.20	18.23	20.62	20.21	17.24	13.16	08.87	04.58	03.19
31.30	32.60	41.10	56.00	67.20	78.00	82.20	80.10	73.10	62.40	47.90	34.90
SAN DIEGO, CA											
32.70											
11.07	14.37	18.52	21.98	22.73	23.40	24.81	23.35	19.49	15.59	12.06	10.26
64.60	65.60	66.00	67.60	69.40	71.10	75.30	77.30	76.50	73.80	70.10	66.10
SAN FRANCISCO, CA											
37.60											
08.03	11.45	16.51	21.79	25.26	26.98	27.14	24.02	19.77	13.92	09.32	07.29
55.30	58.60	61.00	63.50	66.60	70.20	70.90	71.60	73.60	70.30	63.30	56.30
SAN JUAN, PR											
18.40											
15.04	17.43	20.29	21.46	20.58	20.62	21.27	20.86	19.00	17.20	15.52	14.03
81.90	82.10	83.60	84.40	85.60	87.00	87.00	87.50	87.60	87.40	85.00	83.10
SAULT STE MARIE, MI											
46.50											
03.69	06.85	11.67	15.70	19.16	20.55	20.83	17.28	11.91	07.64	03.77	02.87
22.00	23.70	32.50	47.20	59.40	70.00	75.10	73.40	64.50	54.80	39.00	26.80
SAVANNAH, GA											
32.10											
09.01	11.85	15.87	20.00	21.02	20.93	20.24	18.40	15.48	13.81	10.68	08.54
61.10	63.60	69.50	77.80	84.80	89.30	90.80	90.30	85.40	78.20	69.30	62.10
SEATTLE, WA											
47.40											
02.97	05.62	09.63	14.68	19.45	20.45	25.52	18.34	13.03	07.45	03.83	02.40
43.40	48.50	51.50	57.00	64.10	69.00	75.10	73.80	68.70	59.40	50.40	45.40
ST. LOUIS, MO											
38.70											
07.12	10.05	13.67	17.75	21.24	23.75	23.26	20.62	16.56	12.48	08.15	06.02
39.90	44.20	53.00	67.00	76.00	84.90	88.40	87.20	80.10	69.80	54.10	42.70
TAMPA, FL											
28.00											
11.47	14.29	18.09	21.66	22.68	20.97	19.89	18.76	16.93	15.28	12.57	10.62
70.60	71.90	76.10	82.40	87.50	89.90	90.10	90.40	89.00	83.90	77.10	72.00
TRAVERSE CITY, MI											
44.70											
03.53	06.44	11.36	15.95	19.62	21.70	21.67	18.26	13.23	08.55	04.28	02.91
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
WAKE ISLAND, PI											
19.30											
15.31	17.84	20.56	22.18	23.32	23.22	21.84	21.24	19.73	17.83	16.31	14.88
81.60	81.70	82.50	83.10	84.60	86.30	87.10	87.80	87.60	86.60	84.90	83.10

WASHINGTON, DC
38.90

06.49	09.25	12.77	16.56	19.50	21.57	20.63	18.36	15.21	11.39	07.39	05.46
41.20	43.40	52.70	65.00	74.50	82.70	86.40	85.00	78.70	68.20	55.60	43.30

WEST PALM BEACH, FL
26.70

11.35	13.99	17.66	20.59	20.93	19.37	20.19	18.88	16.10	13.89	12.03	10.88
75.00	76.00	79.30	82.90	86.10	88.30	89.60	90.20	88.30	84.30	79.50	76.10

WHIDBEY ISLAND, WA
48.30

03.21	06.04	10.42	15.27	19.98	20.66	22.48	18.07	13.32	07.43	04.05	02.64
00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00

WILMINGTON, DE
39.70

06.49	09.38	13.04	16.80	19.41	21.37	20.69	18.32	14.96	11.17	07.32	05.55
40.20	42.20	51.10	63.00	73.10	81.60	85.50	83.90	78.20	67.80	55.20	43.00

YAKIMA, WA
46.60

04.14	07.56	12.74	18.13	22.79	24.61	26.76	22.41	16.83	10.11	05.04	03.35
36.40	46.10	54.80	64.10	73.10	79.70	88.10	85.90	78.30	64.70	48.50	39.10

YAKUTAT, AK
59.50

01.14	03.01	07.07	11.93	14.38	15.26	13.70	10.69	07.20	03.91	01.54	00.58
31.20	35.00	37.90	43.60	50.80	56.40	59.30	59.60	55.40	47.30	38.20	32.70

TMATRIX.DAT

The data contain in the Tmatrix data file is also used, as a correction factor, in the computations performed when determining the Solar Array and Battery sizes. It consists of an array of 24 numeric values as follows:

-021.34	-013.54	-002.06	009.94	018.98	023.35
021.60	014.17	002.97	-008.94	-018.64	-023.22
447.15	443.66	437.22	430.47	423.15	419.08
418.79	422.14	428.36	435.69	442.67	446.77

SolrCalc.Bas Program Listing

```
1000 REM ——————————————————————————————————————————————
1010 REM
1020 REM        Program   :   SolrCalc.Bas
1025 REM        Revision  :   5.0
1030 REM        Author    :   Melvin C. Wiley (UNISYS)
1040 REM
1050 REM ——————————————————————————————————————————————
1060 DIM F(24, 5), Q(24, 1), C$(100), S$(100)
1070 DIM L(100), B2(12, 100), B(12, 2), H1(12, 2)
1080 DIM B1(12, 100), D(365), U(10), G(10), Y(10)
1090 DIM Z(10), A(12), H(12, 2), W(365), S(24, 4), FLAG(10), OT$(20)
1100 REM DIM HDR1$, HDR2$, OT$(20)
1110 REM DIM R1, R2, R3, R4, R5, R6, R7, R8, R9, R10
1120 REM DIM R11, R12, R13, R14, R15, R16, R17, R18, R19, R20
1130 REM DIM R21, R22, R23, R24, R25, R26, R27, R28, R29, R30
1140 REM DIM R31, R32, R33, R34, R35, R36, R37, R38, R39, R40
1150 PI = 3.1415927#
1160 RV$ = CHR$(255) + 'AE'
1170 NV$ = CHR$(255) + 'AA'
1180 DEF FNACS (X) = ATN(SQR(1 - X ^ 2) / X)
1190 DEF FNNACS (X) = PI + FNACS(X)
1200 DEF FNAT$ (CL, RW) = CHR$(255) + 'C' + CHR$(RW) + CHR$(CL)
1210 HDR1$ = 'Solar Calculator - Revision 5.0'
1220 HDR2$ = 'US Coast Guard'
1230 OT$(1) = ' '
1240 OT$(2) = '=============================================================='
1250 OT$(3) = 'Solar Calculator - Revision 5.0  –  Overview'
1260 OT$(4) = '=============================================================='
1270 OT$(5) = ' '
1280 OT$(6) = '  The Solar Calculator is used to calculate the solar panel'
1290 OT$(7) = 'and battery sizes required to maintain a solar power system'
1300 OT$(8) = 'for a period of time based upon the load, available sunlite'
1310 OT$(9) = '(insolation), latitude and tilt angle.                      '
1320 OT$(10) = ' '
1330 OT$(11) = '=============================================================='
1340 R1 = 0:  R2 = 0:   R3 = 0:  R4 = 0:  R5 = 0:  R6 = 0:  R7 = 0
1350 R8 = 0:  R9 = 0:  R10 = 0: R11 = 0: R12 = 0: R13 = 0: R14 = 0
1360 R15 = 0: R16 = 0: R17 = 0: R18 = 0: R19 = 0: R20 = 0: R21 = 0
1370 R22 = 0: R23 = 0: R24 = 0: R25 = 0: R26 = 0: R27 = 0: R28 = 0
1380 R29 = 0: R30 = 0: R31 = 0: R32 = 0: R33 = 0: R34 = 0: R35 = 0
1390 R36 = 0: R37 = 0: R38 = 0: R39 = 0: R40 = 0
1400 FOR I = 0 TO 12
1410     A(I) = 0
1420     B(I, 0) = 0:  B(I, 1) = 0:  B(I, 2) = 0
1430     H(I, 0) = 0:  H(I, 1) = 0:  H(I, 2) = 0
1440     H1(I, 0) = 0: H1(I, 1) = 0: H1(I, 2) = 0
1450 NEXT I
1460 FOR I = 0 TO 10
1470     U(I) = 0: G(I) = 0: Y(I) = 0: Z(I) = 0: : FLAG(I) = 0
1480 NEXT I
1490 FOR I = 0 TO 365
1500     W(I) = 0: D(I) = 0
```

```
1510 NEXT I
1520 FOR I = 0 TO 24
1530     F(I, 0) = 0: F(I, 1) = 0: F(I, 2) = 0
1540     F(I, 3) = 0: F(I, 4) = 0: F(I, 5) = 0
1550     Q(I, 0) = 0: Q(I, 1) = 0
1560     S(I, 0) = 0: S(I, 1) = 0: S(I, 2) = 0
1570     S(I, 3) = 0: S(I, 4) = 0
1580 NEXT I
1590 FOR I = 0 TO 100
1600     C$(I) = "": S$(I) = "": L(I) = 0
1610     B1(0, I) = 0: B2(0, 1) = 0
1620     B1(1, I) = 0: B2(1, 1) = 0:  B1(2, I) = 0: B2(2, 1) = 0
1630     B1(3, I) = 0: B2(3, 1) = 0:  B1(4, I) = 0: B2(4, 1) = 0
1640     B1(5, I) = 0: B2(5, 1) = 0:  B1(6, I) = 0: B2(6, 1) = 0
1650     B1(7, I) = 0: B2(7, 1) = 0:  B1(8, I) = 0: B2(8, 1) = 0
1660     B1(9, I) = 0: B2(9, 1) = 0:  B1(10, I) = 0: B2(10, 1) = 0
1670     B1(11, I) = 0: B2(11, 1) = 0: B1(12, I) = 0: B2(12, 1) = 0
1680 NEXT I
1690 GOSUB 7480
1700 GOSUB 7330
1710 GOSUB 7480
1720 FOR N = 1 TO 10
1730     FLAG(N) = 0
1740 NEXT N
1750 R1 = 1
1760 R2 = 1
1770 PRINT
1780 INPUT " Enter the Aid Number..:"; N$
1790 INPUT "          Aid Name....:"; NME$
1800 PRINT : A$ = ""
1810 INPUT " Are you using standard Coast Guard solar panels (Y/N)..<Y>:"; A$
1820 IF A$ = "" THEN A$ = "Y"
1830 IF A$ = "Y" OR A$ = "y" THEN VP = 13.8: GOTO 1870
1840 VP = 0
1850 INPUT "    Enter the Max Power Point for the panels used....<13.8>:"; VP
1860 IF VP <= 0 THEN VP = 13.8
1870 R2 = 1
1880 R1 = 1
1890 A$ = ""
1900 INPUT " Do you know what the Array Size will be (Y/N)..........<N>:"; A$
1910 IF A$ = "" THEN A$ = "N"
1920 IF A$ <> "Y" AND A$ <> "y" THEN R2 = 0: GOTO 1970
1930 O = 0
1940 INPUT "    Enter the Array Size (in watts) to be Used.........<10>:"; O
1950 IF O <= 0 THEN O = 10
1960 RR = O / VP
1970 A$ = ""
1980 INPUT " Do you know what the Battery Size will be (Y/N)........<N>:"; A$
1990 IF A$ = "" THEN A$ = "N"
2000 IF A$ <> "Y" AND A$ <> "y" THEN R1 = 0: GOTO 2080
2010 K = 0
2020 INPUT "    Enter the Battery Size (in amp-hrs) to be used....<100>:"; K
2030 IF K <= 0 THEN K = 100
2040 ZZ = K
```

```
2050 C = 0
2060 INPUT '   Enter the Initial Battery Capacity (in amp-hrs)...<100>:'; C
2070 IF C <= 0 THEN C = 100
2080 IF R1 + R2 = 2 THEN FLAG(3) = 1: GOTO 2110
2090 IF R1 + R2 = 0 THEN FLAG(6) = 1: GOTO 2110
2100 IF R1 = 0 THEN FLAG(7) = 1
2110 X = 0
2120 INPUT ' What is the Duration of Analysis (in years)............<5>:'; X
2130 IF X <= 0 THEN X = 5
2140 X = INT(X)
2150 INSMON = 0
2160 INPUT ' What is the Month of Installation (ie Jan = 1).........<1>:'; INSMON
2170 IF INSMON < 1 THEN INSMON = 1
2180 IF INSMON > 12 THEN INSMON = 12
2190 V = INT((INSMON * 2) - 1)
2200 INSDAY = 0
2210 INPUT ' What is the Day of Installation (1 - 31)...............<1>:'; INSDAY
2220 IF INSDAY < 1 THEN INSDAY = 1
2230 IF INSDAY > 31 THEN INSDAY = 31
2240 IF INSDAY > 15 THEN V = V + 1
2250 IF INSMON = 1 THEN O$ = 'January'
2260 IF INSMON = 2 THEN O$ = 'February'
2270 IF INSMON = 3 THEN O$ = 'March'
2280 IF INSMON = 4 THEN O$ = 'April'
2290 IF INSMON = 5 THEN O$ = 'May'
2300 IF INSMON = 6 THEN O$ = 'June'
2310 IF INSMON = 7 THEN O$ = 'July'
2320 IF INSMON = 8 THEN O$ = 'August'
2330 IF INSMON = 9 THEN O$ = 'September'
2340 IF INSMON = 10 THEN O$ = 'October'
2350 IF INSMON = 11 THEN O$ = 'November'
2360 IF INSMON = 12 THEN O$ = 'December'
2370 IF INSDAY < 16 THEN O$ = O$ + ' (first half)'
2380 IF INSDAY > 15 THEN O$ = O$ + ' (second half)'
2390 A$ = ''
2400 INPUT ' Is the Insolation Data stored on the Disk (Y/N)........<Y>:'; A$
2410 IF A$ = '' THEN A$ = 'Y'
2420 IF A$ <> 'Y' AND A$ <> 'y' THEN J = 0: GOSUB 6430: GOTO 2700
2430 J = 0
2440 INPUT '   Enter the Data File Number..........................<1>:'; J
2450 IF J < 1 THEN J = 1
2460 OPEN 'I', #1, 'SOLRDATA.DAT'
2470 FOR R3 = 1 TO J
2480    INPUT #1, C$(R3), S$(R3)
2490    INPUT #1, XL$
2500    L(R3) = VAL(XL$)
2510    FOR I = 1 TO 12
2520       INPUT #1, XB1$
2530       B1(I, R3) = VAL(XB1$)
2540       B(I, 1) = B1(I, R3)
2550    NEXT I
```

```
2560    FOR KK = 1 TO 12
2570       INPUT #1, XB2$
2580       B2(KK, R3) = VAL(XB2$)
2590       B(KK, 2) = B2(KK, R3)
2600    NEXT KK
2610    L1 = L(R3)
2620 NEXT R3
2630 R3 = 0
2640 CLOSE #1
2650 IF FLAG(2) = 1 THEN 2780
2660 TEMPLAT = 0
2670 PRINT USING ' Enter the Data Sites Latitude....................< ##.# >:'; L1;
2680 INPUT TEMPLAT
2690 IF TEMPLAT > 0 THEN L1 = TEMPLAT
2700 IF E$ = 'Y' THEN 2780
2710 E = 0
2720 INPUT ' Enter the array Tilt from horizontal (in Degrees).....<60>:'; E
2730 IF E < 1 THEN E = 60
2740 IF E > 89 THEN E = 89
2750 AA = PI / 180
2760 L2 = L1 * AA
2770 E1 = E * AA
2780 IF F$ = 'N' THEN 3200
2790 GOSUB 7480
2800 PRINT
2810 R4 = 0
2820 INPUT ' Enter the number of electrical loads to be supported...<1>:'; R4
2830 IF R4 < 1 THEN R4 = 1
2840 IF R4 > 10 THEN R4 = 10
2850 LPCTR = 0
2860 FOR N = 1 TO R4
2870    PRINT
2880    PRINT USING '     Electrical Load <##> -> Avg Current Drain (amps) <.55>'; N;
2890    U(N) = 0
2900    INPUT U(N)
2910    IF U(N) <= 0 THEN U(N) = .55
2920    Y(N) = 0
2930    INPUT '                          Duty Cycle (e.g. 10%=10) <10>'; Y(N)
2940    IF Y(N) > 99 THEN Y(N) = 99
2950    IF Y(N) < 1 THEN Y(N) = 10
2960    YY = Y(N) / 100
2970    A$ = ""
2980    INPUT '                       On During Nighttime Only (Y/N) <Y>'; A$
2990    IF A$ = "" THEN A$ = 'Y'
3000    IF A$ = 'Y' OR A$ = 'y' THEN 3090
3010    G(N) = 0
3020    INPUT '                            Number of Hours On/Day <24>'; G(N)
3030    IF G(N) < 1 THEN G(N) = 24
3040    IF G(N) > 24 THEN G(N) = 24
3050    Z(N) = 1
3060    R5 = U(N) * YY * G(N)
3070    R6 = R5 + R6
3080    GOTO 3120
```

```
3090    Z(N) = 0
3100    R7 = U(N) * YY
3110    R8 = R7 + R8
3120    A$ = ": INPUT '                CG-181 Flasher use Required (Y/N) <Y>'; A$
3130    IF A$ = " THEN A$ = 'Y'
3140    IF A$ = 'Y' THEN R9 = YY: R10 = R9 + R10
3150    LPCTR = LPCTR + 1
3160    IF LPCTR = 4 THEN GOSUB 7480: LPCTR = 0
3170 NEXT N
3180 GOSUB 7480
3190 REM ---------------------------------------------------------------
3200 REM                          INPUT PARAMETERS
3210 REM ---------------------------------------------------------------
3220 CTY$ = 'N/A'
3230 STA$ = 'N/A'
3240 IF J <> 0 THEN CTY$ = C$(J): STA$ = S$(J)
3250 PRINT
3260 PRINT FNAT$(6, 2); : PRINT 'INPUT PARAMETERS ARE: ';
3270 PRINT FNAT$(8, 4); : PRINT 'Aid Number (and Name).....: '; N$; '('; NME$; ')';
3280 PRINT FNAT$(9, 4); : PRINT USING 'Month/Day of Installation.: ##/##'; INSMON; INSDAY;
3290 PRINT FNAT$(10, 4); : PRINT USING 'Tilt Angle from Horizontal: ## deg'; E;
3300 PRINT FNAT$(9, 46); : PRINT USING 'Length of Analysis: ## yrs'; X;
3310 PRINT FNAT$(10, 46); : PRINT USING 'Aid Pos - Lat/Long: ##.#N/###.#W'; L1; L9;
3320 PRINT FNAT$(11, 4);
3330 IF R2 = 0 THEN PRINT 'Array Size................: Calc';
3340 IF R2 = 1 THEN PRINT USING 'Array Size................: ##.# watts'; O;
3350 PRINT FNAT$(11, 46);
3360 IF R1 = 0 THEN PRINT 'Battery Size......: Calc';
3370 IF R1 = 1 THEN PRINT USING 'Battery Size......: ####.# AH'; ZZ;
3380 PRINT
3390 GOSUB 4070
3400 GOSUB 4560
3410 IF FLAG(4) = 1 THEN 3520
3420 M = 1
3430 FOR N = 1 TO 12
3440    S(M, 1) = 23.89 * B(N, 1)
3450    S(M + 1, 1) = S(M, 1)
3460    M = M + 2
3470 NEXT N
3480 FOR M = 1 TO 24
3490    Q(M, 1) = S(M, 1)
3500 NEXT M
3510 REM ---------------------------------------------------------------
3520 REM                          SUBROUTINE SIZING
3530 REM ---------------------------------------------------------------
3540 IF FLAG(3) = 1 THEN GOSUB 4880: GOSUB 5000: GOTO 3630
3550 IF FLAG(7) = 1 THEN GOSUB 4880: GOTO 6790
3560 IF FLAG(6) = 1 THEN GOSUB 6630: GOSUB 4880: GOTO 6790
3570 GOSUB 6630
3580 GOSUB 4880
3590 GOSUB 5000
3600 IF FLAG(5) = 1 THEN 3580
3610 FLAG(5) = 0
```

```
3620 REM ————————————————————————————————————————————————
3630 REM                      PRT RESULTS
3640 REM ————————————————————————————————————————————————
3650 IF FLAG(3) = 0 THEN 3660
3660 REM IF R20 < 48 THEN 7020
3670 REM IF R20 > 52 THEN 7100
3680 GOTO 3730
3690 IF O >= 10 THEN 3730
3700 FLAG(3) = 0: FLAG(6) = 0: FLAG(7) = 1
3710 R1 = 0: R2 = 1: O = 10: RR = O / VP
3720 GOTO 3320
3730 PRINT FNAT$(13, 2); : PRINT 'RUN RESULTS ARE:';
3740 PRINT ' ( '*' indicates Non-Standard Configuration )';
3750 BSIZ% = 0: IF INT(ZZ) <> (INT(ZZ / 100) * 100) THEN BSIZ% = 1
3760 MCAP% = 0: IF R20 < 48 THEN MCAP% = 1
3770 IF R20 > 52 THEN MCAP% = 2
3780 PRINT FNAT$(15, 4); :  PRINT USING 'Array Size........: ####.## watts'; O;
3790 PRINT FNAT$(16, 4); :  PRINT USING 'Battery Size......: ####.## AH'; ZZ;
3800 PRINT FNAT$(17, 4); :  PRINT USING 'Min Battery Cap...:  ###.## %'; R20;
3810 PRINT FNAT$(15, 46); : PRINT USING 'Load Amp-Hrs (Min): ###.## AH'; R14;
3820 PRINT FNAT$(16, 46); : PRINT USING '             (Avg): ###.## AH'; R16;
3830 PRINT FNAT$(17, 46); : PRINT USING '             (Max): ###.## AH'; R15;
3840 SPNL10 = 0: SPNL20 = 0: SPNL35 = 0: NUMBAT = 0
3850 NUMBAT = INT(ZZ / 100)
3860 IF NUMBAT < (ZZ / 100) THEN NUMBAT = NUMBAT + 1
3870 WATTS = O
3880 WHILE WATTS >= 35: SPNL35 = SPNL35 + 1: WATTS = WATTS - 35: WEND
3890 WHILE WATTS >= 20: SPNL20 = SPNL20 + 1: WATTS = WATTS - 20: WEND
3900 WHILE WATTS >= 10: SPNL10 = SPNL10 + 1: WATTS = WATTS - 10: WEND
3910 ASIZ% = 0
3920 IF WATTS > 0 THEN SPNL10 = SPNL10 + 1: ASIZ% = 1
3930 PRINT FNAT$(18, 4); :  PRINT USING 'Solar Panel Qty (10 watt): ##'; SPNL10;
3940 PRINT FNAT$(18, 46); : PRINT USING 'Solar Batteries (100AH)..: ##'; NUMBAT
3950 PRINT FNAT$(19, 4); :  PRINT USING '                (20 watt): ##'; SPNL20;
3960 PRINT FNAT$(19, 4); :  PRINT '(suggested)';
3970 PRINT FNAT$(19, 46); : PRINT '(suggested)';
3980 PRINT FNAT$(20, 4); :  PRINT USING '                (35 watt): ##'; SPNL35;
3990 GOSUB 7550
4000 PRINT FNAT$(26, 2); : PRINT '';
4010 A$ = '': INPUT ' Another Run (Y/N) <Y>'; A$
4020 IF A$ = '' THEN A$ = 'Y'
4030 IF A$ = 'Y' OR A$ = 'y' THEN 1210
4040 GOSUB 7480
4050 PRINT ' Solar Calculator Terminated.....': PRINT
4060 END
4070 REM ————————————————————————————————————————————————
4080 REM                    SUBROUTINE DAY
4090 REM ————————————————————————————————————————————————
4100 N = 1
4110 CC = 23.45
4120 BB = .0578
```

```
4130 REM ————————————————————————————————————————
4140 REM                    SUBROUTINE START
4150 REM ————————————————————————————————————————
4160 IF N > 80 THEN 4190
4170 D(N) = CC * SIN(AA * 1.008 * (N - 80))
4180 GOTO 4230
4190 IF N > 266 THEN 4220
4200 D(N) = CC * SIN(AA * .965 * (N - 80))
4210 GOTO 4230
4220 D(N) = (-CC) * SIN(AA * .975 * (N - 266))
4230 REM ————————————————————————————————————————
4240 REM                    SUBROUTINE CORRECTION
4250 REM ————————————————————————————————————————
4260 IF D(N) = 0 THEN D(N) = .000001
4270 R11 = TAN(L2) * TAN(D(N) * AA)
4280 IF R11 >= 1 THEN 4370
4290 IF R11 <= -1 THEN 4390
4300 IF R11 < 0 THEN 4340
4310 QQ = BB / COS(L2) / COS(D(N) * AA) / SIN(FNACS(R11))
4320 HH = (2 / 15) * (FNNACS(-R11) / AA) + 2 * QQ
4330 GOTO 4360
4340 QQ = BB / COS(L2) / COS(D(N) * AA) / SIN(FNNACS(R11))
4350 HH = (2 / 15) * (FNACS(-R11) / AA) + 2 * QQ
4360 IF HH < 24 THEN D(N) = HH: GOTO 4400
4370 D(N) = 24
4380 GOTO 4400
4390 D(N) = 0
4400 N = N + 1
4410 IF N <= 365 THEN 4160
4420 M = 1
4430 I = 1
4440 N = 1
4450 T = 0
4460 T = D(N) + T
4470 N = N + 1
4480 I = I + 1
4490 IF I <= 15 THEN 4460
4500 S(M, 2) = T
4510 I = 1
4520 M = M + 1
4530 T = 0
4540 IF M <= 24 THEN 4460
4550 RETURN
4560 REM ————————————————————————————————————————
4570 REM                    SUBROUTINE LOAD
4580 REM ————————————————————————————————————————
4590 M = 0
4600 N = 1
4610 R12 = 0
4620 R15 = 0
4630 IF R10 <> 0 THEN 4660
4640 T = 0
4650 GOTO 4670
4660 T = (10 + 20 * R10) * (24 - D(N)) / 1000 + .006 * D(N)
```

```
4670 W(N) = T + R6 + R8 * (24 - D(N))
4680 IF N = 1 THEN R14 = W(1)
4690 IF N = 80 THEN 4710
4700 IF W(N) < R14 THEN R14 = W(N)
4710 IF R15 < W(N) THEN R15 = W(N)
4720 R13 = W(N) + R13
4730 R12 = 1 + R12
4740 IF R12 <> 15 THEN 4790
4750 M = M + 1
4760 S(M, 3) = R13
4770 R12 = 0
4780 R13 = 0
4790 N = N + 1
4800 IF N > 365 THEN 4820
4810 GOTO 4630
4820 R16 = 0
4830 FOR N = 1 TO 24
4840    R16 = S(N, 3) + R16
4850 NEXT N
4860 R16 = R16 / 360
4870 RETURN
4880 REM ------------------------------------------------------------------
4890 REM                    SUBROUTINE OUTPUT
4900 REM ------------------------------------------------------------------
4910 DD = 1 / 85.97999
4920 O = VP * RR
4930 IF FLAG(1) = 1 THEN 4950
4940 GOSUB 5570
4950 FOR M = 1 TO 24
4960    S(M, 4) = 15 * (DD * .9199999 * S(M, 1) * RR)
4970 NEXT M
4980 FLAG(1) = 0
4990 RETURN
5000 REM ------------------------------------------------------------------
5010 REM                    SUBROUTINE BATTERY
5020 REM ------------------------------------------------------------------
5030 R19 = V
5040 TT = C
5050 P = C
5060 FF = C
5070 N = 15 * (V - 1)
5080 FLAG(5) = 0
5090 R20 = 100
5100 FOR GG = 1 TO X
5110    IF GG <= 1 THEN 5140
5120    R19 = 1
5130    N = 0
5140    IF V <= 1 THEN 5180
5150    FOR M = 1 TO V
5160       F(M, GG) = 100
5170    NEXT M
```

```
5180    FOR M = R19 TO 24
5190       FOR I = 1 TO 15
5200          N = N + 1
5210          TT = TT - W(N)
5220          R21 = TT / ZZ
5230          IF R21 > .76 THEN 5260
5240          SS = 1 - .1 * R21
5250          GOTO 5330
5260          IF R21 >= .9199999 THEN 5290
5270          SS = 1.266 - .44 * R21
5280          GOTO 5330
5290          IF R21 > .9950001 THEN 5320
5300          SS = 2.31 - 1.58 * R21
5310          GOTO 5330
5320          SS = 0
5330          R22 = S(M, 4) * SS / 15
5340          TT = TT + R22
5350          IF TT <= ZZ THEN 5370
5360          TT = ZZ
5370          IF TT < 0 THEN TT = 0
5380          IF TT < P THEN P = TT
5390       NEXT I
5400       F(M, GG) = P * 100 / ZZ
5410       P = TT
5420       IF R20 > F(M, GG) THEN R20 = F(M, GG)
5430    NEXT M
5440 NEXT GG
5450 IF FLAG(3) = 1 THEN GOTO 5560
5460 IF FLAG(7) = 1 THEN GOTO 5560
5470 IF FLAG(6) = 1 THEN GOTO 5560
5480 IF R20 < 48 THEN 5510
5490 IF R20 > 52 THEN 5530
5500 GOTO 5560
5510 RR = 1.1 * RR
5520 GOTO 5540
5530 RR = .9 * RR
5540 O = VP * RR
5550 FLAG(5) = 1
5560 RETURN
5570 REM ------------------------------------------------------------
5580 REM                         SUBROUTINE TILT
5590 REM ------------------------------------------------------------
5600 IF FLAG(8) = 1 THEN 5730
5610 OPEN "I", #1, "TMATRIX.DAT"
5620 FOR I = 1 TO 12
5630    INPUT #1, XH1$
5640    H1(I, 1) = VAL(XH1$)
5650    H(I, 1) = H1(I, 1)
5660 NEXT I
5670 FOR JJ = 1 TO 12
5680    INPUT #1, XH2$
5690    H1(JJ, 2) = VAL(XH2$)
5700    H(JJ, 2) = H1(JJ, 2)
5710 NEXT JJ
5720 CLOSE #1
```

```
5730 FOR M = 1 TO 24
5740     S(M, 1) = Q(M, 1)
5750 NEXT M
5760 M = 1
5770 FOR N = 1 TO 12
5780     H2 = H(N, 1) * AA
5790     R23 = -TAN(L2) * TAN(H2)
5800     IF R23 <= 1 THEN 5830
5810     R23 = 1
5820     GOTO 5850
5830     IF R23 < -1 THEN R23 = -1
5840     IF R23 < 0 THEN R24 = FNNACS(R23): GOTO 5860
5850     R24 = FNACS(R23)
5860     R25 = -TAN(L2 - E1) * TAN(H2)
5870     IF R25 < 0 THEN R26 = FNNACS(R25): GOTO 5890
5880     R26 = FNACS(R25)
5890     IF R26 >= R24 THEN GOSUB 6250
5900     IF R26 < R24 THEN GOSUB 6340
5910     R27 = .5 * (1 + COS(E1))
5920     R28 = .2 * (1 - COS(E1))
5930     IF R26 >= R24 THEN GOSUB 6090
5940     IF R26 < R24 THEN GOSUB 6170
5950     R29 = S(2 * N - 1, 1) / (.2713 * R30)
5960     IF R29 > .2 THEN 5980
5970     R31 = COS(128.57 * R29 * AA)
5980     IF R29 > .76 THEN R31 = .16: GOTO 6000
5990     R31 = 1.379179 - 3.465989 * R29 + 2.912088 * R29 ^ 2
6000     IF R24 <> 0 THEN 6040
6010     S(M + 1, 1) = 0
6020     S(M, 1) = 0
6030     GOTO 6060
6040     S(M + 1, 1) = S(2 * N - 1, 1) * ((1 - R31) * R32 + R27 * R31 + R28)
6050     S(M, 1) = S(2 * N - 1, 1) * ((1 - R31) * R32 + R27 * R31 + R28)
6060     M = M + 2
6070 NEXT N
6080 RETURN
6090 REM
6100 REM                          SUBROUTINE RD2
6110 REM
6120 IF R24 <> 0 THEN 6150
6130 R30 = 1
6140 GOTO 6160
6150 R30 = (24 / PI) * H(N, 2) * (COS(L2) * COS(H2) * SIN(R24) + (R24) * SIN(L2) * SIN(H2))
6160 RETURN
6170 REM
6180 REM                          SUBROUTINE RD3
6190 REM
6200 IF R24 <> 0 THEN 6230
6210 R30 = 1
6220 GOTO 6240
6230 R30 = (24 / PI) * H(N, 2) * (COS(L2) * COS(H2) * SIN(R26) + (R26) * SIN(L2) * SIN(H2))
6240 RETURN
```

```
6250 REM ----------------------------------------------------------------
6260 REM                          SUBROUTINE RD0
6270 REM ----------------------------------------------------------------
6280 IF R24 <> 0 THEN 6310
6290 R32 = 1
6300 GOTO 6320
6310 R32 = COS(AA*(L1-E))*(SIN(R24)-(R24)*COS(R26))/COS(L2)/(SIN(R24)-(R24)*COS(R24))
6320 FLAG(8) = 1
6330 RETURN
6340 REM ----------------------------------------------------------------
6350 REM                          SUBROUTINE RD1
6360 REM ----------------------------------------------------------------
6370 IF R24 <> 0 THEN 6400
6380 R32 = 1
6390 GOTO 6410
6400 R32 = COS(AA*(L1-E))*(SIN(R26)-(R26)*COS(R26))/COS(L2)/(SIN(R24)-(R24)*COS(R24))
6410 FLAG(8) = 1
6420 RETURN
6430 REM ----------------------------------------------------------------
6440 REM                          SUBROUTINE NEWDATA
6450 REM ----------------------------------------------------------------
6460 J = 0: N = 1
6470 GOSUB 7480: PRINT
6480 TEMPLAT = 0: L1 = 45
6490 PRINT USING ' Enter the Data Sites Latitude..........................< ##.# >:'; L1;
6500 INPUT TEMPLAT
6510 IF TEMPLAT > 0 THEN L1 = TEMPLAT
6520 FOR M = 1 TO 12
6530    PRINT USING ' Enter the Avg Daily Insolation (in Langleys) for Month [##] <334>'; M;
6540    A(M) = 0: INPUT ' '; A(M)
6550    IF A(M) <= 0 THEN A(M) = 334
6560    S(N, 1) = A(M): S(N + 1, 1) = A(M)
6570    Q(N, 1) = A(M): Q(N + 1, 1) = A(M)
6580    N = N + 2
6590 NEXT M
6600 FLAG(4) = 1
6610 GOSUB 7480
6620 RETURN
6630 REM ----------------------------------------------------------------
6640 REM                          SUBROUTINE ARRAY SIZE
6650 REM ----------------------------------------------------------------
6660 FF = 0
6670 P = 0
6680 FLAG(1) = 1
6690 GOSUB 5570
6700 DD = 1 / 85.97999
6710 FOR M = 1 TO 24
6720    FF = S(M, 3) + FF
6730    P = S(M, 1) + P
6740 NEXT M
6750 P = 15 * DD * .9199999 * .9 * P
6760 RR = FF / P
6770 O = VP * RR
6780 RETURN
```

```
6790 REM ----------------------------------------------------------------
6800 REM                         SUB BATTERY SIZE
6810 REM ----------------------------------------------------------------
6820 N = 0
6830 II = 0
6840 Q1 = 0
6850 ZZ = 0
6860 R33 = 0
6870 FOR M = 1 TO 24
6880    N = .9 * S(M, 4) - S(M, 3)
6890    IF N >= 0 THEN 6930
6900    II = N + II
6910    R33 = 1 + R33
6920    GOTO 6940
6930    IF II < Q1 THEN Q1 = II: II = 0
6940 NEXT M
6950 IF R33 = 24 THEN 7130
6960 IF Q1 = 0 THEN GOTO 7210
6970 IF ZZ <> 0 THEN GOTO 7000
6980 ZZ = 2 * ABS(Q1)
6990 GOTO 6870
7000 C = 2.1 * ABS(Q1)
7010 ZZ = C
7020 Q1 = 0
7030 GOSUB 5000
7040 IF R20 < 48 THEN 7070
7050 IF R20 > 52 THEN 7100
7060 GOTO 3630
7070 C = 1.1 * ZZ
7080 ZZ = C
7090 GOTO 7030
7100 C = .9 * ZZ
7110 ZZ = C
7120 GOTO 7030
7130 PRINT
7140 PRINT SPC(13); '------------------------------------------------'
7150 PRINT SPC(13); 'Error:  The Array you have entered is Insufficient'
7160 PRINT USING '              to maintain the battery's charge above ##%.'; R20
7170 PRINT
7180 PRINT SPC(13); '      Re-Run using a Larger Solar Array...'
7190 PRINT SPC(13); '------------------------------------------------'
7200 GOTO 4000
7210 PRINT
7220 PRINT SPC(13); '------------------------------------------------'
7230 PRINT SPC(13); 'Error:  The Array you have entered is Excessive and'
7240 PRINT SPC(13); '      will NEVER allow the battery's charge to fall'
7250 PRINT USING '            below ##%.'; R20
7260 PRINT
7270 PRINT SPC(13); '      Re-Run using a Smaller Solar Array...'
7280 PRINT SPC(13); '------------------------------------------------'
7290 GOTO 4000
```

```
7300 REM
7310 REM                         Program Explanation
7320 REM
7330 DH$ = ""
7340 PRINT
7350 INPUT " Would you like a brief Overview of the SolrCalc (Y/N) <N>"; DH$
7360 IF DH$ = "" THEN DH$ = "N"
7370 IF DH$ <> "Y" AND DH$ <> "y" THEN GOTO 7440
7380 GOSUB 7480
7390 FOR OE% = 1 TO 11
7400    PRINT TAB((80 - LEN(OT$(OE%))) / 2); OT$(OE%)
7410 NEXT OE%
7420 PRINT : PRINT SPC(26);
7430 INPUT "Press Return to Continue"; PRTC$
7440 RETURN
7450 REM
7460 REM                         Program Header
7470 REM
7480 PRINT CHR$(12);
7490 PRINT
7500 PRINT SPC(1); STRING$(78, "-")
7510 PRINT SPC((80 - LEN(HDR1$)) / 2); HDR1$
7520 PRINT SPC((80 - LEN(HDR2$)) / 2); HDR2$
7530 PRINT SPC(1); STRING$(78, "-")
7540 RETURN
7550 REM
7560 REM                         Warning Messages
7570 REM
7580 IF ASIZ% = 1 THEN PRINT FNAT$(15, 3); : PRINT "*";
7590 IF BSIZ% = 1 THEN PRINT FNAT$(16, 3); : PRINT "*";
7600 IF MCAP% <> 1 THEN 7660
7610 PRINT FNAT$(22, 4);
7620 PRINT RV$ + "Warning:  The Array/Battery Size specified will:";
7630 PRINT FNAT$(23, 4);
7640 PRINT "        Cause the battery to discharge below 48%." + NV$;
7650 GOTO 7710
7660 IF MCAP% <> 2 THEN 7710
7670 PRINT FNAT$(22, 4);
7680 PRINT RV$ + "Warning:  The Array/Battery Size specified will:";
7690 PRINT FNAT$(23, 4);
7700 PRINT "        Never let the battery discharge below 52%." + NV$;
7710 RETURN
```